☐ Generate Collection

L8: Entry 5 of 17 File: USPT May 29, 2001

DOCUMENT-IDENTIFIER: US 6240090 B1
TITLE: Self-configuring processors in an asynchronous transfer mode switch

Abstract Text (1):
A self-configuring node includes a number of data processors associated with corresponding ports of an asynchronous transfer mode (ATM) switch. When the node is placed into service, each of the processors automatically broadcasts an initial message to all of the ATM switch ports including each processor's identification and ATM switch port location. After receiving the initial message broadcast by the processor, a designated master processor stores the broadcasting processor's identification and ATM switch port location in a database and sends an acknowledgment directed specifically to the processor broadcasting the initial message. From that received acknowledgment signal, the processor recognizes the identity and ATM switch location of the master processor. Internal Control Paths (ICPs) are established through the ATM switch between processors using the identification and location information stored for each board processor. The internal control paths are used to communicate control messages and other information between the processors. Thus, automatic configuration of a multiprocessor, ATM switch-based node is achieved without requiring polling of the processors by the master processor or involving a human operator.

Application Filing Date (1):
19980428

Brief Summary Text (11):
A self-configuring node includes plural function module boards, each having one or more board processors and a corresponding ATM switch port, connected to available slots of an asynchronous transfer mode (ATM) switch. When the node is placed into service, each of the board processors automatically broadcasts an initial message to all of the ATM switch port locations. The initial message includes each board processor's identification and ATM switch port location.

DOCUMENT-IDENTIFIER: US 6247016 B1
TITLE: Decision tree classifier with integrated building and pruning phases

Application Filing Date (1):
19981110

Brief Summary Text (15):
FIG. 2 is an example of a decision tree for the training data in FIG. 1. Each internal node of the decision tree (denoted by a circle in FIG. 2) has a "test" involving an attribute, and an outgoing branch for each possible outcome. For example, at the root node 10 the test is "is the salary level of the applicant less than $20,000.00?" If the answer to this inquiry is "no," the loan application is automatically accepted, ending the inquiry and establishing a "leaf" 20 (a leaf is the ultimate conclusion of a partition after no further inquiry is to be made, and is denoted by a square in FIG. 2) for the acceptance. Thus, in the example, an applicant who has a salary greater than $20,000 is classified in a class for those applicants who qualify for a loan based on their salary alone.

Brief Summary Text (22):
Each node of the decision tree maintains a separate list for every attribute. Each attribute list contains a single entry for every record in the partition for the node. The attribute list entry for a record contains three fields--the value for the attribute in the record, the class label for the record, and the record identifier. Attribute lists for the root node are constructed at the start using the input data, while for other nodes they are derived from their parent's attribute lists when the parent nodes are split. Attribute lists for numeric attributes at the root node are sorted initially and this sort order is preserved for other nodes by the splitting procedure. Also a histogram is maintained at each node that captures the class distribution of the records at the node. Thus, the initialization of the root node in Step 1 of the build algorithm of FIG. 3 involves (1) constructing the attribute lists, (2) sorting the attribute lists for numeric attributes, and (3) constructing the histogram for the class distribution.

First Hit

☐ Generate Collection

L10: Entry 13 of 36 File: PGPB Jun 14, 2001

DOCUMENT-IDENTIFIER: US 20010003846 A1
TITLE: Encapsulated, streaming media automation and distribution system

Abstract Paragraph:
Disclosed are systems and methods for creating and distributing programming content carried by a digital streaming media to be a plurality of remote nodes located over a large geographic area to create customized broadcast quality programming at the remote nodes. At the remote nodes, a multi-window screen display simultaneously shows different programming including national programming and local programming content. The remote nodes utilize a remote channel origination device to assemble the customized programming at the remote location that can be controlled from a central location. An encapsulated IP and IP encryption system is used to transport the digital streaming media to the appropriate remote nodes. Also disclosed is a graphical user interface ("GUI") providing a software control interface for creating and editing shows or programs that can be aired or played on a remote display device having a multi-window display. The intuitive GUI Software provides the user the ability to easily manage and assemble a series of images, animations and transitions as a single broadcast quality program to be displayed on a remote display device. Another application software system is capable of automating the production of audio narration reports. The disclosed audio concatenation engine automates the creation of audio narration using prerecorded audio segments to minimize the requirement for live, on-air personnel to record audio narration segments.

Application Filing Date:
20001201

Summary of Invention Paragraph:
[0001] This invention relates in general to the automation and distribution of programming information including video, audio, text, and graphics to a large number of program viewers located over a large geographic area. More particularly, it is directed to an integrated, automated production and distribution system for providing customized delivery of digital streaming media to particular geographic areas, markets, groups and/or individuals via remotely controlled origination nodes.

Summary of Invention Paragraph:
[0016] These existing systems address basic requirements such as the ability to reduce the digital bandwidth necessary to carry a video signal of any given quality, the ability to transport these digitally compressed video and audio signals via standard digital transmission and modulation systems whether satellite, fiber, wireless and/or Internet based, and the ability to scramble and control individual authorization of groups and/or specific satellite receivers over a point-to-point and/or multi-point system via the use of key based conditional access and encryption technologies. In most cases, the network and/or regional/sectional programming is distributed from a network headend facility directly to the appropriate redistribution headend. In other words, TV networks are not designed to forward the appropriate program elements both real-time and stored to an automated, remote origination node for customized production, coordination and distribution of broadcast quality localized programming via terrestrial TV,

cable MSO, DTH headend, internet web servers and/or home based processing unit

Summary of Invention Paragraph:
[0020] The exemplary embodiments of the present invention provide an integrated streaming media system capable of generating and distributing broadcast quality streaming media content to a large number of remote nodes located over a large geographic area. In the exemplary embodiments, the network automation and integration may extend beyond the production and generation facilities to extend the capability of centrally scheduled network control to remote locations, if necessary, where programming content can be specifically customized for the particular remote location and/or region. The exemplary embodiments described herein are numerous and have many different aspects and embodiments, any of which may be practiced by alone or in combination with other aspects of the invention.

Summary of Invention Paragraph:
[0021] According to an exemplary embodiment, the streaming media generation and distribution system includes a broadcast or Network Operations Center, a digital distribution system, and Remote Channel Origination Nodes. The Network Operations Center operates 24 hours a day, 7 days a week and houses the broadcast, production, technical and programming operations of the network. From a wide variety of information sources, the Network Operations Center creates the digital streaming media program content carried by a digital streaming media encapsulated by the IP for distribution to the remote nodes over the satellite network. Preferably, the facility will support the acquisition of programming and information to create the live programming for distribution via encapsulated IP transport techniques.

Summary of Invention Paragraph:
[0023] The network operation center preferably includes a Network Automation and Integration subsystem and Network Monitor Distribution and Control subsystem with specialized computer automated and networked components to digitally assemble the programming components to implement the multi-window program display. The Network Automation and Integration preferably manages the "multi-channel" origination to the remote nodes and coordinates with automated production systems to create the different segments for the multi-part screen with the individualized audio narratives. The facility preferably includes multiple production areas to enable the concurrent production of regional and local weather segments. The automated production systems preferably mange the incoming information, such as weather data, and digitally distribute it to the different production areas for reformatting and editing.

Summary of Invention Paragraph:
[0027] These interactive/transactional components are carried and distributed by a digital transmission system utilizing encapsulated IP techniques. This method allows the appropriate interactive applications and data elements to be distributed to a specific, predetermined headend via a specific, predetermined remote node. In this way, an effective overlay, virtual private network ("VPN") is constructed to deliver the appropriate interactive components to the appropriate headend device through a network Remote Channel Origination Node ("RCON") via vendor specified physical interfaces to digital tier, cable headend equipment.

Summary of Invention Paragraph:
[0029] The exemplary network operations center preferably supports and synchronizes local insertion at the remote nodes to offer "local avail" opportunities. For example, the system will be developed to support the delivery of localized weather information with audio and data, where each of the remote nodes receiving specific weather data and commercials. The network will preferably deploy insertion capabilities at the remote nodes to enable insertion of local weather programming in addition to local advertising. Preferably, the insertion capability is controlled from the network operations center using a Graphical User Interface application software to enable an operator to integrate, create, edit and control

the streaming media content. The Graphical User Interface software provides a system operator with a means to remotely control the streaming media as it is generated by the network RCON for final distribution. Preferably, the software enables the network automation and integration subsystem to synchronize the timing and display of various national and local programming information at the various receiving nodes as is described in more detail herein.

Summary of Invention Paragraph:
[0030] The network distribution and management will enable each of the remote nodes to receive program content and generate customized programming. In this embodiment, the remote content origination nodes are capable of creating the local programming from the received program content according to a program schedule that is centrally generated from the channel, traffic & contract management subsystem internal to the network operations center. Local commercial and/or program insertion is integrated by standard cueing techniques such as use of contact closures and/or tone based switching as well as central scheduling of local playback or insertion facilities..

Summary of Invention Paragraph:
[0032] The disclosed embodiments provide many new features and advantages for generating, integrating and distributing information content and customized programming over a large geographic area. The automated subsystems at the network operations center provides for greater levels of network integration and enables an all-digital production facility allowing greater levels of integration and automation to enable more efficient operation of the system with fewer personnel headcount resulting in substantial savings to the system operator. The digital streaming media and IP techniques allows all information, video, audio, data and control information to be distributed through a common digital format. The remote nodes located at the remote locations throughout the geographic area allows locally customized programming to be assembled from national program components and information components assembled with local data or locally produced programming. Control of the programming at the remote nodes may be controlled from the NOC system using a variety application software programs and hadware including networked client-server workstations, file servers, databases, etc.

Summary of Invention Paragraph:
[0033] A Graphical User Interface (GUI) software application program enables the scheduling, configuration, monitoring and control of programming content from a central location. The GUI provides a convenient mechanism for an operator to control, cue and synchronize the programming displayed at remote node. The control and synchronization information can be transmitted to the remote nodes as a component of the multiplexed digital streaming media. A plurality of Graphical User Interface processes can control the multi-window screen display at the remote locations.

Summary of Invention Paragraph:
[0034] The disclosed embodiments provide a number of advantages in implementing a digital transmission system. The automated subsystems provide integration that enables the system operator to achieve efficiency and operational cost savings in delivering customized broadcast quality digital television programming to individual remote locations and nodes. The automated production of audio narration allows the production of voice segments without requiring live on-air personnel. The GUI allows an operator to control, cue and synchronize the programming displayed at remote locations that can be carried by the encapsulated digital streaming media.

Brief Description of Drawings Paragraph:
[0050] FIG. 14 shows a block diagram of a remote channel origination node for streaming, interactive media product;

Detail Description Paragraph:
[0058] FIG. 1 shows a high-level system diagram embodying a streaming media generation and distribution system 50 including streaming media production, distribution, and monitoring and control system employing different aspects of the present invention. Generally, the streaming media distribution system 50 provides programming content that can be distributed over large geographic areas while enabling the capability to provide specific local program content to specific geographic areas, groups and/or individuals. In particular embodiments, interactive streaming media allows viewers to control or select further programming content to be viewed or particular features to be displayed, controlled or selected. In another embodiment, transactional streaming media may allow viewer to select and then conduct a transaction such as but not limited to allowing the viewer to request or provide further information, purchase a product, make a financial wager, manage bank or security accounts, make a contract, process applications and/or control devices.

Detail Description Paragraph:
[0059] The streaming media distribution system 50 of the present embodiment integrates a variety of live, rendered and stored program elements, real-time data with taped and/or stored programming, stored and rendered graphics and live & stored local program content and/or effects to provide customized local programming content, programming elements and system components that can be used to provide broadcast-quality streaming media. Broadcast quality streaming media is preferably that suitable for use in video or television programming. These real-time and store and forward components are distributed from a Network Operations Center 300 and to Remote Channel Origination Nodes ("RCONs") 500 to enable the customized provision of local program content to specific geographic areas, groups and/or individuals. The RCONs 500 are connected to headend devices 450 to distribute programming to viewers.

Detail Description Paragraph:
[0061] For purposes of clarity and ease of description the exemplary network embodiment is shown with regard to the major operational areas identified as: Data Analysis and Graphics 100, Commercial Transaction Processing 200, Network Operations Center 300 and Remote Channel Origination Node 500. It should be understood that the exemplary embodiment may include fewer or additional operational areas, which may be segmented according to different operational or functional areas.

Detail Description Paragraph:
[0064] Generally, Network Operations 300 receives data and information components from the Data Analysis and Graphics 100 such as the weather data, forecast, analysis, graphics and animated sequences. These among other components are used to produce the "ready for air," national network programming feed. The national feed is distributed to Remote Channel Origination Nodes ("RCON") 500 simultaneously with other local programs as well as interactive and/or transaction components and enablers for final rendering of the local channel and distribution to end users via the appropriate headend device. The RCON 500 may include a broadcast distribution system such as a satellite transmission network and a plurality of receiver nodes where each RCON 500 receives only elements meant specifically for that node by way of IP encryption techniques whether in a serial, broadcast and/or hybrid distribution configuration. Several exemplary embodiments are described in more detail in FIGS. 13A-13E.

Detail Description Paragraph:
[0130] Referring now to FIG. 6, shown is the system Network Operations Center ("NOC") 300 that manages, integrates and automates various production, multiplexing and distribution functions of the system. Generally, the NOC 300 implements and utilizes a distribution network which transmits real-time program and data elements along with store and forward components in a digital streaming media via an IP

based Network Distribution, Monitoring and Control System 370 to the various RCONs 500 at remote locations throughout the desired viewing areas. RCON 500 preferably interfaces a plurality of remote nodes including headends 450, which receive streaming media, information products and program elements addressed for that node by way of encapsulated IP, IP encryption and IP addressing techniques whether in a serial, broadcast and/or hybrid distribution configuration. Moreover, RCONs 500 provide the network interface with the downstream headends device 450 so that the interactive application software and real-time data components of the transactional product can be transported to and from the end user's TV computational device which may reside in a set-top box, computer of even within the TV itself. In a preferred embodiment, RCON 500 receives national feed video for programming and also has the capability to assemble local programming and perform commercial insertion using a variety of program elements that may be distributed via the encapsulated IP transport or provided locally. In this embodiment, the headend device 450 is the network facility responsible for final distribution of the streaming media programming to the viewer or end user such as a TV affiliate, a cable system headend, a DTH satellite uplink facility, web server and/or home based processing unit as further described in FIGS. 13A-13E. Further details of the distribution network will be provided herein.

Detail Description Paragraph:
[0137] Network Distribution, Monitor & Control 370 provides the means through which streaming as well as "store & forward" program elements are (1) packaged via IP encapsulation and addressed via IP encryption for distribution to various remote channel origination nodes 500, (2) monitored and verified by way of a "quantum" monitoring system consisting of "cloned" RCONs 500 and alarm and status updates from in the field RCONs 500 and (3) controlled for purposes of pre-emption for weather alerts, restoral of RCON functionality and detailed monitoring of RCON functions. Network Distribution, Monitor & Control 370 is described further in reference to the description of FIG. 15. In this exemplary embodiment, the elements of the streaming media distribution may include IP distribution as the transport "wrapper" of the streaming media including national/common programming elements with associated stored graphic, data, schedules, commands, heuristics and executable software modules is distributed to various remote channel origination nodes. The system may also utilize IP encryption as the means of scrambling distributed program elements on both serial and point to point and/or multi-cast networks.

Detail Description Paragraph:
[0223] The multi-part screen will be a computer generated composite of graphic images created in the graphic production area 3304. The screen will be a series of computer generated graphic templates archived to the integrated network server 3302, and distributed as resident files to each of the RCONs 500. The multi-screen software management system is the application used to assemble the program segment files according to a predetermined order and schedule as developed by the Local Day of Air Scheduling terminal 3518 as depicted in FIG. 10 CT&CM 350 and as fuirther described in FIG. 17. This multi-part screen management package is a software applications that enables control of the programming compiled and displayed at the remote node.

Detail Description Paragraph:
[0226] Set Top Box Application Development & Management 410 can be implemented by various interactive television and set top box development systems such as the OpenTV Operating Environment available from OpenTV of Mountain View, Calif. OpenTV is a complete operating system for digital interactive television systems capable of meeting the demands of the digital broadcast consumer electronics market. OpenTV system includes a Software Development Kit (SDK) providing a complete C-language content development environment for developers creating interactive television with graphical content that has the look and feel of television. The OpenTV SDK allows for interactive applications to be developed using OpenTV's application programming

interface (API) along with graphical and command-line tools. OpenTV also provides a MPEG encoder that converts a variety of different types of images into television ready pictures. The OpenTV encoder can convert images from a variety of file formats such as TIFF, GIF, JPEG, BMP, PICT and PBM and creates professional quality MPEG still pictures suitable for television broadcasting. OpenTV Web provides digital television operators the ability to repurpose HTML content for television displays. OpenTV Web allows televisions that, unlike conventional personal computer browsers, do not have significant amounts of processing power or memory to display web content by shifting the HTML processing load to servers in the broadcasting headend. Using this application content can be reused rather than recreated and the content is compressed for quicker download.

Detail Description Paragraph:
[0235] Remote Channel Origination Node

Detail Description Paragraph:
[0236] Referring now to FIG. 14 shown is the Remote Channel Origination Node 500 which provides the final origination and/or custom rendering of the broadcast quality, interactive product for distributions to end-users and subscribers. In this exemplary embodiment, program elements are distributed over a DVB satellite distribution system via an encapsulated IP transport as depicted in FIG. 14 which includes the national network feed, a plurality of audio channels for regional and local distribution, weather graphics files, local condition data, software, data, graphics, animated sequences, and interactive program components. The satellite distribution system 3704 also carries conditional access, switching cues, site addressable data via IP encryption, CG, bug and rendering files, commands, remote automation system schedules, software upgrades, etc.

Detail Description Paragraph:
[0273] Referring now to FIG. 17, shown is a exemplary embodiment of the graphic user interface ("GUI") of a software control interface for creating and editing shows or programs that can be aired or played on a remote display device having a multiple number of display windows such as shown in FIG. 2. The intuitive GUI Control Software provides the user the ability to easily manage and assemble a series of images, animations and transitions as a single weather show to be displayed in a remote display device. Separate sets of still graphics and animation loops can be maintained and edited to be combined and synchronized to create programs or sequences for the various display windows. Using the GUI Control Software makes it relatively easy to change and edit the sequence from a central control device by moving pictures or icons representing the graphics or animation loops in the GUI with a computer mouse device. The GUI 900 control interface residing at the NOC 300 allows centrally scheduled control of the network and remote nodes.

Detail Description Paragraph:
[0317] At the remote node or headend display devices, RCON 500 receives the control signals in the streaming media to synchronize the programming content to create the desired programming in each of the multiple windows. The particular schedules and control signals are addressed to the RCON according the encrypted IP addressing and conditional access techniques that can be provided by vendors such as Media4. The RCON 500 may be implemented with the Duet Platform from Chyron and developed using Lyric and CAL as previously described.

Detail Description Paragraph:
[0337] The many disclosed embodiments herein allow the creation of a broadcast network system for displaying a multi-window screen programming customized for different remote locations by enabling the generation, distribution, assembly and synchronization of digital streaming media from a network operation center. The network operations center provides an all-digital program facility ~~that~~ creates and broadcasts programming content in a digital streaming media format that is

transmitted to a plurality of remote nodes that can utilize the digital streaming media along with locally provided information to create programming customized for particular remote locations. A number of computer subsystems, workstations, networks, file servers, databases and communication links described herein allow the automation of network operation center processes and functions to integrate and automate operations, resulting in simplified operation and thus lower operating costs requiring less personnel. A graphical user interface provides operator access to a software control program that allows the control of the multi-window display at the remote location from a central location such as the network operation center. The audio concatenation engine provides the operator the ability to automate the creation of audio narration without requiring the traditional number of production studios or on-air personnel.

CLAIMS:

1. An information distribution system delivering digital program information over a large geographic area wherein the digital program information provides different broadcast quality television programming to a plurality of remote locations within the large geographic area, the system comprising: a network operation center creating a national program feed and information components that are multiplexed to create a digital streaming media; a broadcast quality multi-window screen display at the remote location, the multi-window display comprising a plurality of different programming in each of the multi-window screens, wherein at least a portion of the different programming is carried by the digital streaming media from the network operation center and comprises at least a national program feed; a graphical user interface accessing a software control process at the network operation center for controlling through the digital streaming media the assembly of the multi-window screen display; a distribution system transmitting the digital streaming media to a plurality of remote locations; and a plurality of remote nodes receiving the digital streaming media at the plurality of remote locations wherein the digital streaming media is used to produce the multi-window display at the remote node.

2. The invention of claim 2 wherein the remote nodes use the digital streaming media and locally provided information from sources at the remote node location to produce customized programming with local content.

3. The invention of claims 1 wherein the digital streaming media is transmitted to the plurality of remote nodes using an encapsulated Internet Protocol (IP) with IP encryption techniques.

4. The invention of claim 3 wherein the encapsulated IP with IP encryption distributes the digital streaming media to remote nodes configured in a serial network configuration.

6. The invention of claim 3 wherein the encapsulated IP with IP encryption distributes the digital streaming media to remote nodes configured in a point to multi-point network configuration.

8. The invention of claim 1 the network operation center further comprising: an audio concatenation engine producing audio narration that is transmitted to the remote nodes by the digital streaming media.

10. An information distribution system delivering digital program information over a large geographic area wherein the digital program information provides customized programming to a plurality of remote locations within the large geographic area, the system comprising: a network operation center creating a national program feed and information components that are multiplexed to create a digital streaming media; a broadcast quality multi-window television display at the remote location, the multi-window display comprising a plurality of different programming in each of

the multi-window screens, wherein at least a portion of the different programming is carried by the digital streaming media from the network operation center and comprises at least a national program feed; an application software process to create audio narrations by assembling audio narrations from a stored format without requiring a live, on-air announcer to produce the narration; a distribution system transmitting the digital streaming media to a plurality of remote locations; a plurality of remote nodes receiving the digital streaming media at the plurality of remote locations wherein the digital streaming media is used to produce the multi-window display at the remote node; locally provided information relevant to the remote location where a remote node is located, wherein the locally provided information is utilized by the remote node for programming in at least one of the multi-windows.

First Hit Fwd Refs

Generate Collection

L11: Entry 7 of 11 File: USPT Aug 7, 2001

DOCUMENT-IDENTIFIER: US 6272536 B1
TITLE: System and method for the distribution of code and data

Application Filing Date (1):
19990113

Brief Summary Text (16):
In summary, the invention consists of a system and method for the large scale distribution of application code and data. The system consists of a client-side component, which the user uses to download applications, as well as a server-side component, which is used by a content provider or developer to make applications available for distribution.

Other Reference Publication (16):
Bennett, J.M. and Bauer, Michael A., "An Analysis of Replication Strategies for X.500-like Distributed Directories," Proceedings, Workshop On the Management of Replicated Data, Nov. 8-9, 1990, pp 137-42 IEEE Comput. Soc. Press Los Alamitos, CA.

Other Reference Publication (71):
Colton, Malcom, "Replicated Data in a Distributed Environment," Proc. Of the 1993 ACM SIGMOD International Conference on Management of Data, vol. 22, Issue 2, May, 1993, pp. 464-466, ACM Press, Washington, D.C., USA.

Other Reference Publication (95):
Wedde, Horst F. et al, Distributed Management of Replicated and Partitioned Files Under Dragon Slayer, Conference Publication: Compsac90, The Fourteenth Annual International Computer Software and Applications Conference, Oct. 1990, pp. 436-441, Institute of Electrical and Electronics Engineers, Inc., New York, New York, USA.

L5: Entry 9 of 26 File: USPT Nov 6, 2001

DOCUMENT-IDENTIFIER: US 6314460 B1
TITLE: Method and apparatus for analyzing a storage network based on incomplete information from multiple respective controllers

Application Filing Date (1):
19981030

Brief Summary Text (17):
In the preferred embodiment, the storage network analyzer is part of a larger distributed storage management program which supports management of storage networks connected to multiple host computer systems through one or more controllers in each respective host. The distributed storage management program comprises a central manager portion and a separate agent in each of the host computer systems. The agents gather data and communicate with the manager across a communications path which is independent of the storage network. The manager collates the data from different agents to produce a coherent view of the network. The storage network analyzer is part of the agent in the host, specifically, in the preferred embodiment it is part of a function called the network daemon.

Detailed Description Text (108):
While running, the local library is normally idling while waiting for a request from the central manager. When the request is received, the library must parse it to determine how the information will be obtained. There is an action corresponding to each type of request, which may also vary with the parameters of the request. Most information used to satisfy requests is obtained from shared memory 1003, but it may also be obtained from ssaraid facility 1012, ODM facility 1010, or directly from an adapter itself. The requested information might be read directly from data records (e.g., from shared memory), or it might involve an exchange of communications between the library and the adapter. For example, a LL_GetAttr request, which requests attribute information, is handled by calling the ssaraid facility 1012 if the requested attribute is a RAID-related attribute, but the same request is handled by retrieving information in shared memory 1003 if the requested attribute is an event count. The local library must also determine in which data structure and location the information can be found. The requested attribute may, for example, be located in one place for a disk and another for an adapter. Finally, some requests contain "wild cards" which may be used in place of specific parameters of the request.

First Hit

Generate Collection

L10: Entry 11 of 36 File: PGPB Jan 24, 2002

DOCUMENT-IDENTIFIER: US 20020010798 A1
TITLE: Differentiated content and application delivery via internet

Application Filing Date:
20010216

Summary of Invention Paragraph:
[0016] It is often the case that the content delivery and distribution provider has a large number of geographically dispersed content delivery and distribution servers. It could manage to forward the requests to the content delivery and distribution servers, using some form of location based resolution of DNS names to IP addresses, based on the origin of the request. Assuming that the content delivery and distribution servers have the desired content cached or mirrored, are relatively near the client, and are not overloaded, then these objects can be served quickly and transparently. This reduces significantly the latency for content arrival, a critical objective in today's web. It should be noted that in this arrangement, the content providers, which control the origin servers, need know nothing about the distribution policy of the content delivery and distribution provider.

Summary of Invention Paragraph:
[0056] 9. Different site views. This important service allows for different views of a site to different customers. Specifically, when an edge server is located within a possibly secured private client domain, such as an intranet, it can be used as a differentiation mechanism that provides special services for a particular class of clients. Such services may not be provided to clients who are outside the class and who access the origin site directly or who access a shared edge server. This form of service differentiation is particularly useful in business-to-business (B2B) applications, e.g., clients of a market research company, CAD system providers or a large wholesale vendor. This service enables different views of the content at different edge server locations and provides special regional and local services and privileges. Such services are enabled for a particular edge server by the downloading of CDML instruction and data structures from the origin site, the target site or a third party site, and by interpretation of CDML code and data structure within the content. Other features of this service include the following:

Summary of Invention Paragraph:
[0066] 12. Origin site load balancing. In some preferred embodiments the edge server may direct user requests to or download content from unloaded origin or relatively close servers. The edge server may maintain a profile of servers load and network condition by measuring rates and latencies experienced with a set of remote origin servers. Using such an approach, the best origin servers in terms of network distance and loading combination may be selected.

Summary of Invention Paragraph:
[0072] The invention provides a content and application delivery system comprising an origin web site which has an origin web server having a first memory for storing a first version of web content. The system includes an edge server communicating via a data network with the origin web server and with a policy control server. The

edge server has a second memory for storing a second version of the web content and deriving the second version from the origin web server according to directives of a service policy that resides at the policy control server. The edge server downloads the directives of the service policy from the policy control server via the data network. A request of a user directed to the origin web site for a resource from the web content is redirected to the edge server, and responsive to the request a third version of the web content is provided to the user from the edge server. the third version is derived from the second version in accordance with the directives of the service policy.

Summary of Invention Paragraph:
[0125] Another aspect of the invention includes the steps of compressing the first version, downloading the first version from the first server to the second server, uncompressing the first version in the second server, and deriving the second version from the first version in the second server.

Detail Description Paragraph:
[0173] In some preferred embodiments regional servers are assigned to group domain names, using a naming convention such as znn-regionalX, where X again represents some number, by causing the regional server to download membership instructions from the origin server or farm in a manner which is disclosed hereinbelow. Such instructions or configuration files for edge server grouping are managed within the znn.com domain. The regional servers at each region then apply these mappings using a local DNS insertion mechanism.

Detail Description Paragraph:
[0195] In addition to the built-in policies, CDML enables a fully programmable interface through the scripting language, CDSL, that can be used to ship policies, such as the one indicated in the above code fragment, as attribute values attached to nodes.

Detail Description Paragraph:
[0196] CDML is based on Extended Markup Language (XML), i.e., it defines an XML document type definition (DTD), and therefore validity of the document can readily be tested using a standard XML parser. The editor is also syntax-directed, taking advantage of XML. In order to provide flexibility in expressing differentiation while still complying with the DTD, CDML allows any nesting of characteristics to take place. For example, a given URL may be associated with a policy that states that certain regions should download a given URL only after five o'clock p.m. local time. This could be expressed by the following pseudo CDML code:

Detail Description Paragraph:
[0197] In this case time is the major characteristic and location is secondary. An alternative policy could require downloading of a page having a high priority to region A, a page having medium priority to region B, and one having low priority to region C. Here the region is the major characteristic, and content-based differentiation is secondary. Hence, in the XML tree the regional characteristic would contain the temporal characteristic. Finally, CDSL, as well as the built-in semantic actions of the CDML interpreter may be based on an extension to the standard XML style sheet language transformations (XSLT), or externally implemented.

Detail Description Paragraph:
[0223] Once defined, edge server profiles are transferred to the edge servers, such as the edge server 84. While the front edge tool 86 logically controls the profiles on the edge server 84, the physical transfer of profiles to the edge server 84 is done through the origin web site 82. The front edge tool 86 stores the profiles on the origin web site 82, and the edge server 84 downloads its profile from the origin web site 82 via standard client-server web protocols using the Internet 88. The profile editing service is effectively decoupled from the responsibility for

reliable and secure storage and transfer of information, which is conventionally provided by the origin server.

Detail Description Paragraph:
[0228] The edge server periodically accesses the origin site in order to download updated profiles. The minimal interval between downloads is configurable and is generally in the order of a few minutes. Dynamic changes of edge server profiles are thus enabled.

Detail Description Paragraph:
[0309] The system 80 supports the capability to predefine meaningful grouping of edge servers, in order to simplify the application of policies applicable to a set of related edge servers. For example, edge servers may be organized by region, organization, or common interest. The edge server grouping, or edge server maps, are defined using the front edge tool 86. Administrators of the system 80 can construct hierarchies of edge servers, which are later used by profile editors in order to apply a particular service directive to a collection of edge servers. In particular, the front edge tool 86 supports defining service profiles for any internal node in the hierarchy, resulting in the application of the service directive to all edge servers in the node's subtree.

Detail Description Paragraph:
[0311] Edge server grouping is a hierarchy in which the edge servers are leaves, and internal nodes represent logical aggregations of nodes. Multiple hierarchies may exist, and the an edge server may belong to multiple hierarchies.

Detail Description Paragraph:
[0312] The front edge tool 86 supports the following edge server grouping operations: (1) creation of a new edge server group or subgroup; (2) removal of a group or subgroup; (3) addition of an edge server to a group or to a number of groups; deletion of an edge server from a group, or from any number of groups; (5) copying, pasting, and moving a group including all of its subtrees recursively; and (6) renaming an edge server group or node.

Detail Description Paragraph:
[0324] The system 80 has the important capability of dynamically delegating control to the edge server. When a delegation is in force, all user requests that are intercepted by a delegatee edge server are handled completely by the edge server, without involving the origin server, except for possibly downloading profiles from the origin server. Thus, if a resource that was requested by a user resides in the cache, it is served. But if the resource is not in the cache, instead of referring the request to the origin server, an error message is returned to the user. This service is useful when the origin server is experiencing an extremely high load, e.g., a denial of service attack, since it still allows legitimate users to receive partial service, and it reduces the load on the origin server.

Detail Description Paragraph:
[0335] The edge server 84 may download profiles from multiple sources. A specific profile source is denoted as the primary source. The primary source provides the edge server 84 with a list of additional profile sources The primary source may also specify certain restrictions with respect to service profile content that apply to the other sources.

Detail Description Paragraph:
[0411] The NAME field can start with the character `*` as the first node. This is a special pattern that forces a global removal of all the names ending with the postfix of the NAME field (without the `*` character). For example "*.znn.com" means all the names that end with "znn.com" and include "a.b.znn.com", "a.znn.com", "znn.com", etc. It should be noted note that the meaning of the `*` character is slightly different than the one described in the document RFC1035.

Detail Description Paragraph:
File Downloads with Registration/Authorization

Detail Description Paragraph:
[0425] Often there is a requirement for the origin server 10 to allow large file downloads. Such downloads can be software binaries, PDF, MP3 or MPEG files, having sizes in the order of tens to hundreds Mbytes. It is advantageous that such files be available in the cache of the regional edge server 30 and other edge servers, particularly if they are repeatedly downloaded to remote areas, in order to reduce the overhead of the servers of the relatively slow internet backbone.

Detail Description Paragraph:
[0426] It is possible, however, that the content provider permits downloading of such content only after the user requesting the material has been authorized or otherwise properly identified. For example, free software is typically given after a proper form has been completed. In such a case the origin server 10 dynamically links the authorized user, such as the client 14 to the downloaded material after obtaining a registration or authorization. In this situation, it is normally inappropriate for the origin server 10 to remotely cache such content. Yet, since such entries are non-cacheable, the load on the origin server 10 and its latency could increase significantly.

Detail Description Paragraph:
[0427] The regional edge server 30 has two different ways of dealing with the caching of such downloads. First, the download is cached or pre-fetched, using its specific service profile. If all users of the regional edge server 30 are authorized then the content is cached. In case users should still not be authorized two alternatives can be employed.

Detail Description Paragraph:
[0428] 1. An unauthorized client 178 is directed to the origin server 10 where the user fills a form, possibly containing a password, or is otherwise authenticated. The client is then provided with a dynamic page where a value URL=X is displayed as the download entry. The user is also stamped with a cookie identifier, NAME, and a value Y. When the request for X is intercepted by the regional edge server 30, it applies a secure function

Detail Description Paragraph:
[0429] The value Z directs the user to the appropriate download content. At this point the cookie may be erased or nullified. Alternatively, cookies are not employed. In this circumstance, when the unauthorized client 178 first approaches the regional edge server 30 for the download, the regional edge server 30 extends the HTTP header with the client IP address V. The origin server 10 responses with a form that is filled in by the client 178, and then with a URL U. When the client 178 accesses the download link U, the function is applied as

Detail Description Paragraph:
[0430] Instead of forwarding the request to the origin server 10 to fill the form, the form is preferably provided to the unauthorized client 178 by the regional edge server 30. This can be done by downloading a page using JSP or some other in-page code. The code is executed by the regional edge server 30, and if the form is found to be valid, the page that enables the download is sent to the client 178.

Detail Description Paragraph:
[0460] The directives editor 122 allows the user to create any hierarchical group structure, and the structure is dynamic. The user can insert a new level after an existing level of the hierarchy, remove an existing level, move edge servers from node to node, and more.

Detail Description Table CWU:
2TABLE 2 cache A cache is a program's local store of response messages and the subsystem that controls this message storage, retrieval, and deletion. A cache stores cacheable responses in order to reduce the response time and network bandwidth consumption for future, equivalent requests. Edge Server A subordinate software server that resides in an "edge" of the internet (e.g., at an internet service provider) and provides enhanced content delivery serv- ices to users on behalf of one or more origin servers. Edge DNS A modified local DNS server according to the Server disclosure herein. External A domain name zone that does not identity the organi- domain name zations associated with the client or the regional zone DNS. front edge A tool for defining service profiles. tool In-bound/ Inbound and outbound refer to the request and response outbound paths for messages: "inbound" means "traveling toward an origin server", and "outbound" means "traveling toward the user agent". Origin Web The web site on which a given resource resides or is Site to be created. Each origin server is mapped to a singe domain name, but it may be mapped to physical hosts in different ways. In particular, a single origin server may span multiple physical servers, and many origin servers can be co-hosted by the same physical server. Proxy A proxy is an intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible trans- lation, to other servers. A proxy MUST implement both HTT? client and server protocol sides. A "transparent proxy" is a proxy that does not modify the request or response beyond what is required for proxy authentica- tion and identification. A "non-transparent proxy" is a proxy that modifies the request or response in order to provide some added service to the user agent, such as group annotation services, media type transfor- mation, protocol reduction, or anonymity filtering. Resource A resource is a network data object or service that can be identified by a Uniform Resource Identifier (URI) . Resources may be available in multiple repre- sentations (e.g. multiple languages, data formats, size, and resolutions) or may vary in other ways. Service A set of properties that parameterize the application Attributes of a certain service directive by defining the content to which it is applied, at what times it is applied, on what edge server's it is applied, and for which customers. Service An instantiated request for service from one or more directive edge server's. It consists of the type of service and the actual service attributes. A service directive is created using the front edge tool. Service A service profile defines what services an edge server profile should provide and under what conditions each service is enabled. edge server A collection of service, management and configuration profile directives chat apply to a specific edge server. An edge server profile can be derived from multiple pro- files. Target A service attribute that defines the content to which content a certain service directive applies. Target A service attribute that defines the group of customers end-users to which a certain service directive applies. Target edge A service attribute that defines the group of edge server group servers to which a certain service directive applies. Target time A service attribute that defines the times during which a certain service directive applies. User agent The client that initiates a request for a resource. These are often browsers, editors, spiders (web-traversing robots), or other end user tools. Name The process done by the name servers (NSs) of Resolution searching through the DNS to find data for which they are not authoritative. Since the DNS is structured as an inverted tree, a NS needs only the domain names and addresses of the root NSs. BIND Berkeley Internet Name Daemon (BIND) is the most widely used NS in the internet. Currently there are two supported versions: BIND 4.9.3, BIND 8.2.2, and a Beta version of BIND 9. Caching In order to reduce DNS traffic, and delays, an NS has an internal cache that stores valid replies, and also has a negative cache that stores valid errors. A time-to-live value is attached to each cached entry, so it can be invalidated. Normally the cache is inter- nal, and is updated only upon DNS replies to queries. Domain Name The full domain name of any node in the DNS tree is the sequence of labels on the path from the node to the root, with dots separating the names in the path. For example: "mail.versedge.com". The maximum length of a domain name is 255 bytes. Domain Name

An inverted tree of labels (each label is a node and Space is limited by 63 characters). The root of the tree is (empty label name) . Each node can have an unlimited number of children. The tree depth is limited to 127 levels. Forwarding NS An NS that forwards a query to a predefined NS if it can't find the data locally in the cache or in the authoritative data. In case of forwarding, the request is always recursive. Fallover Sending a DNS query to the authoritative DNS server when there is no valid cached information. Master Server An ultimate source of information about a domain. A primary master is an authoritative server configured to be the source of zone transfer for one or more secondary servers. Name Server The program that stores information about the DNS. Name servers generally have complete information about some part of the DNS, called a zone. The name server is than said to have authority for that zone. Zones usually represent administrative boundaries. A DNS server is authoritative for an external domain name zone provided that the DNS server is defined as such at the root DNS or at another authoritative server for the external domain name zone. Recursion The action a name server performs when it receives a recursive query. In recursion, the NS must reply with the requested data or with an error if data was not found. In order to complete the Recursion, the name server itself can use a recursive query to other name server, or use an Iteration method. Resolver a resolver is the client of the DNS system, who accesses name servers. The resolver is used by local programs. The resolver handles: -Querying a name server -Interpreting a response from the name server - Returning the information to the program that requested it. Resource The data associated with domain names is contained in Record Resource Records (RR's) . Records are divided into classes. The internet class (IN) is the most popular and default class. Within a class, records come in several types, which correspond to the different varieties of data that may be stored in the domain name space. (Example: Address, Mail exchange (MX), Canonical name (CNAME)) RFC Request For Comments. The Internet Request For Comments (or RFC) documents are the written definitions of the protocols and policies of the Internet. Root NS An NS that is authoritative to the top-level domains, or nows where there are authoritative NS's to a top-level domain. Slave Server A slave server, also called a secondary server, is an authoritative server that uses zone transfers from the primary master server to retrieve the zone data, or optionally from a cache. Subdomain Subdomain Delegation involves assigning responsibility Delegation for some part of the domain to other Authoritative Name Servers. Delegation is achieved by pointing to the authoritative name servers instead of containing information for the subdomain. Zone A zone contains all the domain names and data of a domain, except for domain names and data in delegated subdomains of that domain. If a subdomain of the domain is not the subject of delegation, however, the zone contains the domain names and data in the subdomain. Example: The zone "versedge.com" has a delegated subdomain "test.versedge.com". The zone

Detail Description Table CWU:

4 Listing 2 <?xml version="1.0"?> <!DOCTYPE cdml SYSTEM "CDML.dtd" > <document url-name="news.html"> <temporal time="17:00" action="download"> <regional region="Western Europe"/> <regional region="middle East"/> </terrtporal> </document>

CLAIMS:

1. A content and application delivery system comprising: an origin web site having an origin web server, said origin web server having a first memory for storing a first version of a web content; an edge server communicating via a data network with said origin web server and a policy control server; said edge server having a second memory for storing a second version of said web content and deriving said second version from said origin web server according to directives of a service policy that resides at said policy control server, said edge server downloading said directives of said service policy from said policy control server via said data network; wherein a request of a user directed to said origin web site for a resource from said web content is redirected to said edge server, and responsive to

said request a third version of said web content is provided to the user from said edge server, said third version being derived from said second version in accordance with said directives of said service policy.

66. The system according to claim 62, further comprising the steps of: compressing said first version, downloading said first version from said first server to said second server; uncompressing said first version in said second server; and deriving said second version from said first version in said second server.

First Hit Fwd Refs

☐ Generate Collection

L11: Entry 5 of 11 File: USPT Mar 12, 2002

DOCUMENT-IDENTIFIER: US 6356947 B1
TITLE: Data delivery system

Application Filing Date (1):
19990219

Brief Summary Text (13):
Above all, a data delivery system as claimed by the invention has advantages when a large number of client nodes are required to have access to a large number of different contents. Such a system is difficult to realize using the solutions of the prior art.

Brief Summary Text (15):
It is thereby advantageous above all that the central server nodes also centrally manage the distribution of the contents to the data server nodes. A further optimization of the data delivery system is thereby possible, and the quality of service of the data delivery system and the resources required by it are further reduced.

Detailed Description Text (28):
The control unit LC controls the downloading of data sets stored in the content server nodes CONST1 and CONST2 into the storage unit VDB. Upon receipt of a control message from the central server node AS, it sends a request message to one of the content server nodes, which requests the content server node to load a data set with a specific content into the data server node SS1. This content is specified by the control message received. The data set is then received via the communication network KN1 and the communication unit INT1', and is then stored by the control unit LC in the storage unit VDB. Because this downloading process takes place via the broad-band data network KN1, the loading process takes only a short time, compared to the playback time of the data set through the playback unit VPU.

Detailed Description Text (32):
The control unit SH controls the dynamic assignment of data server nodes to the requesting client nodes and manages the distribution of the contents to the data server nodes SS1 to SS4. The assignment is dynamic, to the extent that it is a function of the current status of the data delivery system DPS. The management of the distribution of the content via the control unit SH is advantageous, but not mandatory, for the operation of the data delivery system DPS.

Detailed Description Text (38):
Via the status data stored in the storage unit SDC, the control unit SH also manages the distribution of the content via the data server nodes SS1 to SS2. If it does not find any data server nodes that meet the criteria of the first or the second group defined above, the control unit SH selects one or more data server nodes and, by sending a corresponding control message to the selected data server node or nodes, initiates the downloading of the data set with the desired content into the data server node or nodes.

Detailed Description Text (41):
If none of the data server nodes SS1 to SS4 meets the criterion of the third group,

the control unit SH causes the selected data server node, during the downloading of the data set with the requested content, to overwrite a data set stored in the storage unit VDB. In this case, the control unit SH first selects a data set that is to be overwritten, because the content of this data set is not frequently requested by client nodes, in comparison to the number of the copies of this data set stored in the data server nodes SS1 to SS4. Then the data server node is selected in which this selected data set is stored. From this fourth group, analogous to the selection process described above, the data server node is selected in which a data set with the requested content is to be loaded.

Detailed Description Text (46):
Because the specified content has not yet been stored in the data server node SS1, the control unit SH sends a message L(CONT) to the data server node SS1, by means of which it requests the latter to download a data set with the specified content. The control unit LC processes the message L(CONT) and sends a message RL(CONT) to the content server node CONTS1, which then sends the data set with the content specified in the message CONT via the communications network KN2 to the data server node SS1, where the data set is stored in the storage unit VDB. As a result, the status data stored in the storage unit SP are changed. The new status data are then sent by the control unit SP with the message STA(SS1) to the central server node AS, where they are stored by the control unit SC in the control unit SDC.

CLAIMS:

6. A method as claimed in claim 1, further including:

controlling downloading of data sets from one or more content server nodes into the data server nodes by the central server node.

First Hit

☐ Generate Collection

L11: Entry 1 of 11 File: PGPB Jun 6, 2002

DOCUMENT-IDENTIFIER: US 20020069420 A1
TITLE: System and process for delivery of content over a network

Abstract Paragraph:
Network systems and processes for delivery of electronic content to recipient processors may be configured to facilitate the delivery of relatively large content items and/or a relatively large number of content items. In one example, a movie rental system and process delivers (downloads) movie files to customers across the Internet. System network architecture includes three or more layers of servers, including a main server supporting an interface for recipient processors to request content items, a plurality of parent servers, and a plurality of edge servers. Edge servers are distributed throughout a region in which the system provides content delivery services to recipient processors. Parent servers, spaced across the service region, support edge servers. A copy of each content item available through the service is distributed from the main server to each parent servers for storage. Parent servers distribute content to edges servers, pursuant to instructions from the main server or requests from the edge servers. The main server receives and processes requests for content items from recipient processors and directs recipient processors to edge servers for obtaining requested content items. The main server also authenticates licenses and publishes rules for the distribution of content items.

Application Filing Date:
20010406

Summary of Invention Paragraph:
[0002] For example, the Internet is used by a variety of business, organizations, governments and individuals to deliver various types of content to Internet enabled devices, including website content for displaying a website on a user's computer, as well as digitally encoded content files. Some Internet websites provide opportunites to download files containing digitally encoded text (such as papers, forms, patents), pictures, images or computer games. In some websites, larger files, such as video files, video games or music files may be downloaded by a user. As a result, a variety of content delivery services are presently provided over communication networks, such as the Internet.

Summary of Invention Paragraph:
[0003] The speed and efficiency at which network users may download content over a network can be largly dependent upon the size of the content files, the number of users simultaneously using the content delivery service and the speed and efficiency of the servers and other system components used by the content delivery service. As the number of users of a content delivery service increases, the delays experienced by users attempting to download content files can increase, unless the service operator provides sufficiently fast and efficent servers and other system components. However, robust servers and system components are expensive to obtain and operate. Thus, with many Internet sites, users may experience significant delays in downloading content files, especially during higher traffic periods and with sites that deliver large content files.

Summary of Invention Paragraph:

[0004] In addition, the delivery of large files can require a large storage capacity and bandwidth, as compared to smaller files. These factors can render conventional systems impractical for downloading large numbers of large files. Thus, typical conventional systems may be impractical or inefficient for providing large-scale services for delivery of large files, such as movie files, music files, video game files or other large program or data files, to users on a network. Accordingly, there is an industry demand for an efficient manner of providing an on-line service for deliverying large numbers of large files, for example, to many users over a wide region.

Summary of Invention Paragraph:
[0008] One example embodiment relates to a movie rental system and process for delivering (downloading) movie files to customers across the Internet. In that example, the content may comprise at least one electronic, digital copy of a movie. Other embodiments of the invention may involve delivery of other types of content including, but not limited to, music files, still image files, game programs, or other software or data.

Summary of Invention Paragraph:
[0011] The main server system supports a website (or other interface) accessable to users on UNDs over a network, such as the Internet. The website may provide an interface for allowing users to select and request content items for downloading. The website may also provide a means for users to purchase a license to access requested content items.

Summary of Invention Paragraph:
[0012] Downloading every content item directly from the main server may require a very large and complex server (especially where the content items are relatively large and/or the number of content items is large). Accordingly, in some embodiments, the function of downloading content items to users is distributed among the edge servers. When a user purchases a license for a selected content item the main server provides the user's UND with a URL (uniform resource locator) that will allow the user to connect to an edge server to download an encrypted electronic file containing the selected content item. The user may input the URL to the user's web browser immediately or wait and download the content item at a later time. Alternatively, this URL can be provided transparently to the user's web browser to cause the download to begin immediately.

Detail Description Paragraph:
[0022] As described above, embodiments of the invention relate to systems and processes for delivery of electronic content to recipient processors over a network. The term "electronic content" (or "content") is used herein to refer to all forms of electronic information (information that may be communicated and processed in an electronic form), including, but not limited to electronic files, streamed data, or other data formats for movies, video, music or other audio, still images, game programs, application software, electronic books, episodic television content or other the like Embodiments of the invention may be configured to facilitate the delivery of a relatively large number of content files, data streams or other data arrangements (and/or one or more relatively large content file, data stream or other data arrangement) over a network to a plurality of recipient devices (or users). In yet further embodiments, a system and process is configured for providing a content delivery service, for delivering such content to many users located over a relatively wide region. For example, the system and process may be used as a mechanism for a content owner or holder to distribute content to users on a network, such as the Internet. However, many aspects of the invention may be used in other contexts, including, but not limited to industrial, military, scientific, educational or other contexts in which the delivery of a large number of files and/or a number of large files to a plurality of recipient devices on a network is desired.

Detail Description Paragraph:
[0024] One example embodiment comprises a movie rental system and process for delivering (downloading) movie files to customers across the Internet. In that example, the content may comprise at least one electronic, digital copy of a movie. Other embodiments of the invention may involve delivery of other types of content including, but not limited to, music files, still image files, game programs, television content, or other software or data.

Detail Description Paragraph:
[0031] According to example embodiments of the invention, the main server 12 interacts with the UNDs 22, to provide a user interface for allowing users to select content items for downloading. In some embodiments, the interface comprises a main website accessable over the network 12 (for example, the Internet), to users on UNDs 22. The main server 12 supports the operation of the interface (for example, website), including receiving information, such as requests or selections of content items, from UNDs 22.

Detail Description Paragraph:
[0033] Copies of the content items (for example, movies) are stored again on the parent servers 14. Additional copies are stored on edge servers 16 throughout a distribution region, such as throughout the United States. A user of a UND interacts with the main server for browsing and conducting commercial transactions. However, the main server redirects the UNDs (transparently to the users, in some embodiments) to edge servers 16 for downloading content items.

Detail Description Paragraph:
[0035] In example embodiments, the user may browse through website text and images describing content items, such as movies, available for downloading. The website may provide the user with search and other functions to assist the user in locating and selecting content items. When the user has found a content item (for example, a movie) that the user would like to access, the user pays a license fee, for example, by submitting payment information, such as credit card, bank card, bank account, pre-paid account or other suitable form of payment information or recordable renumeration. Alternatively, upon the user selecting a content item, a license fee payment may be deducted automatically from such cards, accounts or other payment sources, transparent to the user. The license fee purchases a limited license to access the selected content item (for example, view a selected movie).

Detail Description Paragraph:
[0037] Once the UND has downloaded the selected content item (for example, movie), the UND may operate a media player tool, such as, but not limited to Windows Media Player or Real Player, or other tool for playing or otherwise accessing the downloaded content item (for example, movie). The user may open the downloaded content file (for example, movie file) using the media player tool. When the user opens the content file, the media player tool connects to the website to verify and enable the license. Once the license has been enabled, the user may freely access the content item (for example, view the movie) through the media player tool, based on the policy of the license. In one example embodiment, the license policy may allow the user to access a content item (for example, view a movie) as many times as the user desires during a predefined window of time, such as a 24-hour period, starting from when the license is enabled. After the license expires, the user or another person may access the content item (for example, movie) again by purchasing and downloading a new license, without downloading the content item (for example, movie) again.

Detail Description Paragraph:
[0047] When a user purchases a license for a selected content item (for example, movie), the main website also provides the user's UND with a URL (uniform resource locator) that will allow the user to download the encrypted electronic file containing the content item (for example, movie) from an edge server associated

with the URL. The user may input the URL to the user 's web browser immediately or wait and download the content item (for example, movie) at a later time. Alternatively, this URL can be provided transparently to the user's web browser to cause the download to begin immediately.

Detail Description Paragraph:
[0048] When the user attempts to download the content item (for example, movie), the system authenticates that this user is currently permitted to download the content item before providing the content item to the user. The actual file that the user will access for download need not be stored on the physical server system associated with the main website, but rather on an edge server 16 elsewhere within the network. When the download has been approved, the system will resolve the URL of an appropriate edge server 16 for downloading the encrypted electronic file containing the requested content item (for example, movie) to the user's UND.

Detail Description Paragraph:
[0054] As described above, the main server system 12 supports the main website. The main server system 12 implements business logic for providing functions associated with user interaction, such as browsing, selecting content items, and purchasing or otherwise obtaining licenses for content items (for example, for rental of movies). The main server system 12 also implements the administration tools of the website, tools for authenticating download requests and security functions for enabling licenses to access content items (for example, view movies). The main server system 12 may authenticate download requests to ensure that the requesting user has purchased or otherwise obtained a valid license, is in a geographical region where the content item (for example, movie) can be downloaded (according to licensing agreements or other geographic restrictions that may be associated with the service or selected content), and/or is not repeatedly downloading the content item (for example, in a denial of service attack).

Detail Description Paragraph:
[0063] The main server system 12 also includes a content management system (CMS) server 38, for controlling the distribution of content and instructions to parent servers 14. The control of content items and instructions distributed from the main server 12 to parent servers 14 may be based on instructions received from external sources, such as a source server 18.

Detail Description Paragraph:
[0070] Downloading every content item directly from the main server 12 may require a very large and complex server (especially where the content items are relatively large and/or the number of content items is large). Accordingly, in preferred embodiments, the function of downloading content items (for example, movies) to users is distributed among multiple servers. Conventional caching approaches based on file accesses are not designed for a mix of very large and very small files and could remove inappropriate files from the cache, from the business view of the service operator.

Detail Description Paragraph:
[0072] In operation, content items are first distributed to edge servers to be available for downloading to service customers. Once sufficient distribution has occurred, users may access the system through the interface (for example website) provided by the main server system 12, to select, submit a request and purchase (or otherwise obtain) a license for one or more content items available through the service.

Detail Description Paragraph:
[0081] For example, the edge servers 16 may receive instructions containing lists of "hot" content items (for example, "hot" movies) or other classifications relating to content items that should be stored on the respective edge servers. In one example embodiment, the main server 12 publishes an XML document containing a

list of all content items (for example, movies) available on the main server 12, with some content items marked as "hot". Additional parameters may also be included, such as geographic restrictions (so that a content item, such as a movie, that is not available for download in a region will never be stored on edge servers in that region). In one embodiment, the main server 12 publishes instructions to the parent servers 14 and the edge servers 16 and the parent servers republish instructions to edge servers.

Detail Description Paragraph:
[0087] Once the content items are sufficiently distributed to edge servers 16, users may access the system through the interface (for example, a main website as described above) supported by the main server system 12. By connecting to the main website, users may search and select content items and submit a request to download content items, and purchase (or otherwise obtain) a license for content items.

Detail Description Paragraph:
[0092] In another embodiment, the main server 12 receives data simply identifying a geographical region associated with the user's IP address from the geographic locator system. Then software associated with the main server 12 determines whether or not a user's IP address is within a geographic region to which the requested content may be delivered, without violating contractual obligations or other restrictions. In one example embodiment, geographic restrictions are implemented by the system such that only users in the United States are allowed to download content items (for example, movies).

Detail Description Paragraph:
[0093] To verify that the content item (for example, movie) has not been excessively downloaded using a particular URL, the main server compares the token in the URL to a corresponding record maintained by the main server 12 that tracks the number of times this URL (or token) has been used to attempt to download the content item. The main server 12 may determine whether or not the number of attempts have exceeded a pre-defined threshold. For example, a URL (or token) can be used three times before becoming invalid, to allow for download failures. Other maximum use limits may be employed in other embodiments. After exceeding the limit, the token is disabled and the URL is no longer valid. A URL with a disabled token will not be authenticated by the main server 12.

Detail Description Paragraph:
[0094] Thus, the network may authenticate a request to download a content item (such as a movie), for example, by verifying that: (a) the user has purchased a license for the requested content item, (b) the user is in a geographical region that is allowed to download the requested content item, and (c) the user is not attempting to download the content item more times than is reasonable. If one or more of these checks fail, the main server 12 denies the authentication and instructs the edge server not to proceed with the download of the content item (for example, movie). The user may be displayed a message or otherwise directed to contact a customer representative for assistance in proceeding. If, on the other hand, authentication is successful, the selected content item(s) may be downloaded to the user's UND.

Detail Description Paragraph:
[0101] If the user has not previously purchased a license for the content item (for example, the user received a copy of the content item other than by downloading the content item from the service network), the main website will query whether the user would like to purchase a license. In this manner, distribution of copies of the encrypted content files among users and potential users may be encouraged by the service operator, because the content cannot be accessed (the movies cannot be viewed) without a valid purchased license.

Detail Description Paragraph:

[0121] Facilitating a transfer of a large file across the Internet can be expensive. Service operators may be charged for file transfers by affiliate companies that operate network components associated with the service network. Some of this expense may be reduced by employing a distributed caching system in the network, as described above. This expense may be further reduced by avoiding downloads. By distributing copies of encrypted files containing content items (for example, movies) by means other than download on the service network, the cost of the downloads may be avoided. For example, encrypted content files (such as encrypted movie files) may be distributed through mass or directed mailings, hand-outs at shopping malls or other populous locations, or other suitable distribution scheme. In addition, valuable marketing data may be derived from license purchasing from mass distribution, indicating consumer interests in such content items (for example, movies).

Detail Description Paragraph:

[0124] As described above, various system and process embodiments according to the present present invention may be employed to provide a content owner or holder with a mechanism for allowing users to access the content through a network. While a preferred example model for the system and process is a movie rental system for downloading movie files to customers across the Internet, other embodiments may involve delivery of other types of content including, but not limited to, music files, still image files, game software, or other software or data. Moreover, other embodiments may employ other wide area or local area networks. Thus, the references to movie files and movie information in the above-described representative examples do not limit the intention to that context of use. Aspects of the invention described above may be employed in the delivery, rental, sale or promotion of other types of content over a network.

First Hit

☐ Generate Collection

L10: Entry 7 of 36 File: PGPB Jun 20, 2002

DOCUMENT-IDENTIFIER: US 20020078461 A1
TITLE: Incasting for downloading files on distributed networks

Application Filing Date:
20001214

Summary of Invention Paragraph:
[0007] U.S. Pat. No. 5,926,101 teaches a multi-hop broadcast network of nodes which have a minimum of hardware resources, such as memory and processing power. The network is configured by gathering information concerning which nodes can communicate with each other using flooding with hop counts and parent routing protocols. A partitioned spanning tree is created and node addresses are assigned so that the address of a child node includes as its most significant bits the address of its parent. This allows the address of the node to be used to determine if the node is to process or resend the packet so that the node can make complete packet routing decisions using only its own address.

Summary of Invention Paragraph:
[0010] U.S. Pat. No. 5,802,301 teaches a method for improving load balancing in a file server. The method includes the steps of determining the existence of an overload condition on a storage device having a plurality of retrieval streams, accessing at least one file thereon, selecting a first retrieval stream reading a file, replicating a portion of the file being read by the first retrieval stream onto a second storage device and reading the replicated portion of the file on the second storage device with a retrieval stream capable of accessing the replicated portion of the file. The method enables the dynamic replication of data objects to respond to fluctuating user demand. The method is particularly useful in file servers such as multimedia servers delivering continuously in real time large multimedia files such as movies.

Summary of Invention Paragraph:
[0014] The most frequent use of such a network is for downloading purposes. A client looks up the content list, and wants to download a particular file/content from the network. The existing protocols for this process are extremely simple and can be described in general as follows. The client or a central server searches the list of servers that contain the desired file, and picks one such server (either randomly or according to some priority list maintained by the central server) and establishes a direct connection between the client requesting the down load and the chosen server. This connection is maintained until the entire file has been transferred. The exact implementation might vary from one protocol to another; however, the fact that only one server is picked for the transfer of the entire requested file remains invariant.

Summary of Invention Paragraph:
[0015] The above-mentioned existing protocols suffer from several serious drawbacks, as stated next. Since only one server is picked for the transfer of the entire file (even though there are potentially many servers with the same content), the quality of service becomes totally dependent on the bandwidth and the reliability of the Internet access that the chosen server maintains during the transfer. This poses a serious problem, especially in the case of networks that

primarily comprise of low-performance servers as is the case for Napster and other proposed peer-to-peer networks and the reliability and speed of the host computers cannot be guaranteed. The average available bandwidth could be as low as that of a 28.8K or a 56K modem. Moreover, the connection of the server to the Internet could be dropped in the middle of a download, necessitating another attempt from the beginning. For example, an average MP3 file is around 5 Mega-bytes in length, and it will take around 16-20 minutes to download it over a 56K modem!! If the connection is dropped at any time during this period, then one needs to attempt the download all over again. The issue of choosing the best server among those that have a copy of the requested file is not properly addressed, leading to a further loss in the quality of the service. If the winner is picked randomly then clearly it is not the best choice. Even if the winner is picked based on a pre-sorted list, where servers are ranked according to their average available bandwidth, the resulting scheme would be far from optimal. In particular, even if a server has a higher average bandwidth, since it comprises only a part of the host computer and shares the bandwidth with other competing tasks, the available bandwidth for the download could be drastically low during the time of the transfer. The protocols do not take advantage of the fact that the client could have a much higher available bandwidth than any of the potential servers. For example, even if the client is connected to a high-speed Ethernet, the effective transfer rate for the session could still be as low as that of a modem that the chosen server might be using. Accuracy and integrity of the downloaded file are not usually guaranteed. Since multiple copies of the files are maintained by different servers the issue of the integrity of the downloaded files becomes a serious concern.

Detail Description Paragraph:
[0030] Referring to FIG. 4 a distributed network 110 includes a plurality of hosts 111 and a shared communication channel 112. Each host is coupled to the shared communication channel 112. Each host 111 may act as both a client and a server and uses the distributed network 110, but not all of the hosts need to act as either a client or a server. The downloading process may be called incasting because it can be construed as a reverse of broadcasting. In broadcasting, a file 120 is transmitted to multiple locations generating multiple copies of the file 120. In contrast, in incasting fragments 121 of multiple copies of the file 120 are gathered together to generate a single copy of the file 120. There is a format for creating and storing multiple copies of the files 120 and a protocol to guarantee fast in the sense that it utilizes the maximum available bandwidth for the task and accurate transfer of the requested content/file 120 to a client in the sense that the content of the copied file 120 is the same as that of the stored one. Incasting would constitute the backbone of the distributed network 110.

Detail Description Paragraph:
[0034] The incasting process will work for any existing format for storing files 120 which follows the convention of being byte aligned. Hence, any server can handle a request, where it is asked to transmit blocks of bytes along with start and end indices. For example, a typical request could be for the transmission of M bytes of a file 120 starting at the kth byte. However, for guaranteeing the integrity of the files 120 and for avoiding expensive retransmissions of potentially erroneous downloads, the following format for storing files 120 and partitioning the file 120 into a specified number of segments 121 is recommended. For each segment 121, compute a message digest of the contents using a secure hash function. The message digest basically acts as a unique identifier for the contents of the segment 121 and on reception, can be used to guarantee the integrity of the contents of the segment 121. In order to guarantee authenticity (e.g., the fact that the file 120 was indeed created by the owner), one can in addition sign the digest. Thus, if one has the segment 121, the message digest and the digital signature of the file 120, then one can verify authenticity (check that the signature matches the digest) and then check for integrity (i.e., the digest matches the contents of the segment 21). For example, the Secure Hash Standard (SHS) can be used to generate 160-bit message digests for the segments 121. The

Digital Signature Standard (DSS) can then be used to generate a 320-bit digital signature of the digest. Other standard hash functions (e.g., MD4 and MD5) and digital signature schemes (e.g., those based on RSA) can be used as well. The number of segments 21 and their starting locations can be stored in the file description. Moreover, if the feature of digital signature is used, then the public key(s) of the owner of the file 20 and the hash function used should also be made available in the description of the files 120.

Detail Description Paragraph:

[0035] Referring to FIG. 6 each entry for a file 120 in a global list 130 contains all the necessary information about the file 120 so that a client can successfully complete an incasting process. The client wishing to download a file 120 goes through the following step of searching the distributed network 110. The client first searches the global list(s) 130 of content/files 120 (to be referred to as the network directory from hereon) to determine the availability of the desired file 120 on the distributed network 110. It is not necessary that a global network directory be maintained at one or several servers. The network directory could itself be maintained in a distributed fashion (e.g., the scheme adopted in the Gnutella network) in which case, a distributed search for the desired content/file 120 will be carried out. In both cases, the following information is returned to the client. A list of (IP) addresses for the servers where the file 120 is located partially or in full. If a server has only parts of the desired file 120, then a succinct description (e.g., start and end byte numbers of contiguous portions of the file 120) of the content stored in the server is also included. If the file 120 is divided into segments 21 along with corresponding digest and digital signature, then the client will also receive descriptions of the segments 21, and the types of hash functions and public key(s) used for the digital signature. The client now has all the storage information about the desired file 20, but does not know the exact availability of bandwidth at the eligible servers for any download request. Using an adaptive incasting algorithm the client is able to virtually segments the file 120 into a number of distinct parts and requests each part from a distinct server. The exact nature of the virtual segmentation procedure will depend on a number of factors, including, the bandwidth available to the client, any prior knowledge about the bandwidth available to different servers and also the storage format of the requested file 120. Since, these are all very implementation-dependent, specific details of the virtual segmentation procedure are not provided. Different servers will respond at different time intervals to the above-mentioned requests. For example, the servers that have high available bandwidth will respond faster than those with slower access, and some servers might not respond at all. The client can then have an online estimate of the traffic and can change the frequency and size of the requests adaptively. Some servers that do not respond during a pre-specified time interval could dropped from the list altogether or could be tried again after an interval of time, if the other active servers are not fast enough. This scheme allows complete flexibility and can be used to saturate the available bandwidth of the client. As the above-mentioned adaptive protocol is carried out, the desired file 20 is received in contiguous chunks of bytes. Since the segmentation format of the file 120 is known to the client, it can always check whether any complete segment 21 of the file 120 has been downloaded or not. Once a full segment 121 of the file 120 is downloaded, it can first verify authenticity of the message digest using the digital signature and the public key and then verify the accuracy/integrity of the segment 121 by comparing the downloaded message digest with a digest that it computes on the content of the segment 121 (using a pre-specified hash function). If any of these verification procedures fails, then it discards the whole segment 121 and starts the requests for the bytes in that segment 121 again. Clearly, there is a tradeoff here between the number of original segments 121 in the file 120 and the number of bytes that might be downloaded multiple times. If there are more segments 121 in the file 120, then first the chance that a segment 121 is corrupted is small, and second even if some bytes are corrupted then only a small number of bytes will need to be downloaded again. However, more segments 121 would mean a larger overhead in terms of the total size

of the file 120. For example, if the Digital Signature Standard is used, then each segment 121 has to have at least an additional 60 bytes: 160 bits (20 bytes) for the message digest and 320 bits (40 bytes) for the digital signature.

Detail Description Paragraph:
[0036] Incasting allows a client to efficiently download a file 120 from the distributed network 110 by putting together fragments of the file 120 obtained from different servers that maintain partial or complete copies of the desired file 120. While the well-known broadcasting procedure creates copies of the same file 120 at many different destination servers incasting recreates a copy of the file 120 by optimally piecing together fragments of the file 120 obtained from multiple target servers. Incasting provides both a suitable format for storing the files 120 and a protocol for gathering the distributed content to create an accurate copy. The same content/file 120 can reside in several different servers on the distributed network 110. This could be either because, the file 120 was created at only one server, and then distributed to several others, or because the same content was created or procured independently at different servers. In fact, our invention will work even if no individual server has the complete file 120, but as long as the complete file 120 is collectively available on the whole distributed network 110. There is a unique identification tag for each content or file 120 residing on the network. A list of all accessible content/files 120 is either available from one central server, or is maintained in a distributed manner (i.e., several servers contain the complete or partial lists of the contents). Such a list would contain the identification tags of all the contents, and for each content/file 120 it would list all the servers that contain a copy of the file 120.

Detail Description Paragraph:
[0037] The most frequent use of the distributed network 10 is for downloading purposes. A client looks up the content list, and wants to download a particular content/file 20 from the distributed network 10. The existing protocols for this process are extremely simple, and can be described in general as follows. The client or a central server searches the list of servers that contain the desired file 20 and picks one such server (either randomly or according to some priority list maintained by the central server) and establishes a direct connection between the client requesting the down load and the chosen server. This connection is maintained until the entire file 20 has been transferred. The exact implementation might vary from one protocol to another; however, the fact that only one server is picked for the transfer of the entire requested file 120 remains invariant.

Detail Description Paragraph:
[0039] From the foregoing it can be seen that incasting for downloading files 120 on distributed networks 110 has been described.

First Hit Fwd Refs

☐ Generate Collection

L10: Entry 18 of 36 File: USPT Jun 25, 2002

DOCUMENT-IDENTIFIER: US 6411616 B1
TITLE: High bandwidth broadcast system having localized multicast access to broadcast content

Application Filing Date (1):
20000419

Detailed Description Text (3):
At each interconnection node is a device called a router, designated here as R1-R6. The function of the router is to receive an input packet of information, examine its source and destination address, and determine the optimal output port for the message. These receive, route determinations, and transmit functions are central to all routers.

Detailed Description Text (46):
One advantage of the foregoing system architecture is that it provides a scaleable architecture that may be sealed to deliver a small number of megabits as well as further scaled to deliver nearly a gigabit of content to a large number of host computers. This architecture is only constrained by satellite transponder capacity, which is typically about 30 mbs per transponder.

Detailed Description Text (94):
3. Software Download

Detailed Description Text (95):
The controller unit 440 handles software downloads for itself and for all of the transponder units 445. Software downloads are preferably performed using FTP file downloads over the local ISP LAN the 240 through NIC 467, from a remote station over the modem interface 470, or through the RS-232 port 487. Before a file is downloaded, FTP server software in the controller unit 440 verifies that the download is, in fact, a new file. The files are preferably downloaded into a fixed directory structure.

Detailed Description Text (97):
The NOC 472 maintains a series of tables used to configure a network of systems such as the one shown in FIG. 15, each system being linked to the NOC 472. These tables may be downloaded using FTP or a predetermined table download command and are used by the controller unit 440 to configure all of the transponder units 445 and to handle any data rate adaptation required by the system. The tables include a Channel Definition Table (CDT), a Carrier Table (CT), and a Channel Cluster Table (CC).

Detailed Description Text (125):
The controlling unit software supports multiple forms of self-diagnostics. Some of the diagnostics run on power up to verify system integrity, and other diagnostic functions are run periodically while the controller unit 440 is operational. For example, the controller unit 440 initially runs several diagnostics including a memory test, a virus scan, a File Allocation Table (FAT) check, a backplane LAN 532 connectivity test, and an external 100 based-T LAN 240 interface test when power is first supplied. As part of its ongoing monitoring process, the controller unit 440

also performs hard drive 455 integrity tests to verify that the file system has not been corrupted. If a hard drive error is encountered, the controller unit 440 logs the error into its trace history, and tries to correct the problem via downloading any corrupted files from the Network Operation Center 472. Still further, the controller unit 440 monitors the fault status of every transponder unit 445 with which it is associated in the respective IPMS 120. The fault monitoring status is an on-going periodic process. All faults are preferably entered into a trace buffer that is available for history tracking. Each fault will be time-stamped and stored in non-volatile memory.

Detailed Description Text (134):
3. Table Download

Detailed Description Text (135):
The network provisioning tables are downloaded via a table download facility. This command is used to process all new tables and reconfigures the system as necessary. The tables are described above.

First Hit

☐ Generate Collection

L10: Entry 6 of 36 File: PGPB Jun 27, 2002

DOCUMENT-IDENTIFIER: US 20020083148 A1
TITLE: System and method for sender initiated caching of personalized content

Application Filing Date:
20001221

Detail Description Paragraph:
[0017] FIG. 1 is a schematic diagram of a preferred embodiment network 10 for sender initiated caching of content. As shown network 10 preferably comprises a core network 16. The core network 16 is the backbone network and preferably comprises one or more routers 18 for routing data packets between Service Providers, for example Internet Service Providers (ISPs) 20, and content provider server 12. One or more users 40 access information utilizing one or more access modes, such as access over copper lines 30, cable access 32, access over optical lines 36, broadband wireless access 34, xDSL access 38 and/or the like. These access modes are preferably broadband access modes which have the ability to deliver large volumes of digital content to the user. Users 40 can access one or more central offices, local exchange, or access nodes 24 utilizing one or more of the above mentioned access modes. In the preferred embodiment, each central office 24 has an associated cache 22 for storing content. In the preferred embodiment, the content stored at cache 22 is personalized broadband content. Central office 24 is communicatively connected to ISP 20 which communicates with one or more routers 18. One or more content provider servers 12 communicate with one or more ISPs 14. ISP 14 communicates with one or more routers 18 of the core network 16. One or more aggregation modules 26 may be utilized to seamlessly aggregate the various access modes for providing access to the central office 24 especially for users utilizing different types of access nodes. ISP 14 provides content from content provider server 12 to core network 16.

Detail Description Paragraph:
[0030] Referring now to FIG. 7, which shows a simplified flowchart of an embodiment of a dynamic web caching on-line session 190 according to the teachings of the present invention. The users each first downloads the online session software application from either edge cache engine 146 or game servers 140 and 134, as shown in block 192. The terms online media session or gaming session are herein defined as any online interactive communication session over a computer network in which one or more users enters input based on displayed content, and content is in turn displayed based on user input. One of the users then launches the game server and invites other users using protocols such as SIP (Session Initiation Protocol) and on-demand multicasting, as shown in blocks 194 and 196. For example, as part of the initiation process the user sends an SIP INVITE with SDP (Session Description Protocol) for exchanging user capability, which includes codes for audio and video UDP ports for voice, video and gaming control messages, as well as for exchanging user profile, which includes information about the user's game personality, tendencies, preferences, etc. The user's profile can be obtained by using user questionnaires or by interactively studying the user's inputs or interaction in response to certain gaming situations. The gaming engine may include a behavior monitor (not shown), which monitors the end user's behavior or activities during a communication or gaming session. The behavior monitor may detect and record how the user plays or behaves in an interactive communication session or game, such as the

user's movement input, type of weapon, hand-and-eye coordination, aiming accuracy, reaction time, skill level, aggressiveness, etc.

First Hit

☐ Generate Collection

L10: Entry 3 of 36 File: PGPB Aug 29, 2002

DOCUMENT-IDENTIFIER: US 20020118638 A1
TITLE: High bandwidth broadcast system having localized multicast access to broadcast content

Application Filing Date:
20010131

Detail Description Paragraph:
[0079] At each interconnection node is a device called a router, designated here as R1-R6. The function of the router is to receive an input packet of information, examine its source and destination address, and determine the optimal output port for the message. These receive, route determinations, and transmit functions are central to all routers.

Detail Description Paragraph:
[0122] One advantage of the foregoing system architecture is that it provides a scaleable architecture that may be scaled to deliver a small number of megabits as well as further scaled to deliver nearly a gigabit of content to a large number of host computers. This architecture is only constrained by satellite transponder capacity, which is typically about 30 mbs per transponder.

Detail Description Paragraph:
[0171] 3. Software Download

Detail Description Paragraph:
[0172] The controller unit 440 handles software downloads for itself and for all of the transponder units 445: Software downloads are preferably performed using FTP file downloads over the local ISP LAN the 240 through NIC 467, from a remote station over the modem interface 470, or through the RS-232 port 487. Before a file is downloaded, FTP server software in the controller unit 440 verifies that the download is, in fact, a new file. The files are preferably downloaded into a fixed directory structure.

Detail Description Paragraph:
[0174] The NOC 472 maintains a series of tables used to configure a network of systems such as the one shown in FIG. 15, each system being linked to the NOC 472. These tables may be downloaded using FTP or a predetermined table download command and are used by the controller unit 440 to configure all of the transponder units 445 and to handle any data rate adaptation required by the system. The tables include a Channel Definition Table (CDT), a Carrier Table (CT), and a Channel Cluster Table (CC).

Detail Description Paragraph:
[0214] The controlling unit software supports multiple forms of self-diagnostics. Some of the diagnostics run on power up to verify system integrity, and other diagnostic functions are run periodically while the controller unit 440 is operational. For example, the controller unit 440 initially runs several diagnostics including a memory test, a virus scan, a File Allocation Table (FAT) check, a backplane LAN 532 connectivity test, and an external 100 based-T LAN 240 interface test when power is first supplied. As part of its ongoing monitoring

First Hit

☐ Generate Collection

L10: Entry 2 of 36 File: PGPB Sep 5, 2002

DOCUMENT-IDENTIFIER: US 20020124098 A1
TITLE: Streaming media subscription mechanism for a content delivery network

Abstract Paragraph:
A reflector network is used within a content delivery network to enable requesting end users to subscribe to live streams that have been published to CDN entry points. A reflector is a generalized packet router program. The reflector network preferably comprises a hierarchy of reflectors that are located at the various entry points into the CDN, at each edge node at which requesting users may be directed by the CDN to obtain live streams, and at various "reflector" nodes located within at least one intermediate layer (in the hierarchy) between the entry points and the edge nodes. The edge nodes and each reflector node also include a manager program that arranges for feeds. When an end user is directed to an edge node that is not yet receiving the desired stream, the edge node's manager issues a subscription request to a set of reflector nodes. If the reflector node(s) are already receiving the desired stream, their reflector(s) begin sending it to the requesting edge node. If, however, the reflector node(s) are not already receiving the desired stream, their manager programs issue the subscription request to the entry point(s) to start the feed.

Application Filing Date:
20010103

Summary of Invention Paragraph:
[0005] Streaming media is a type of Internet content that has the important characteristic of being able to be played while still in the process of being downloaded. A client can play the first packet of the stream, and decompress the second, while receiving the third. Thus, an end user can start enjoying the multimedia without waiting to the end of transmission. Streaming is very useful for delivering media because media files tend to be large particularly as the duration of the programming increases. Indeed, for live events, the file size is, in effect, infinite. To view a media file that is not streamed, users must first download the file to a local hard disk-which may take minutes or even hours-and then open the file with player software that is compatible with the file format. To view streaming media, the user's browser opens player software, which buffers the file for a few seconds and then plays the file while simultaneously downloading it. Unlike software downloads, streaming media files are not stored locally on a user's hard disk. Once the bits representing content are used, the player typically discards them.

Summary of Invention Paragraph:
[0009] is well-known to deliver streaming media using a content delivery network (CDN). A CDN is a self-organizing network of geographically distributed content delivery nodes that are arranged for efficient delivery of digital content (e.g., Web content, streaming media and applications) on behalf of third party content providers. A request from a requesting end user for given content is directed to a "best" replica, where "best" usually means that the item is served to the client quickly compared to the time it would take to fetch it from the content provider origin server.

Summary of Invention Paragraph:
[0013] A reflector network is used in conjunction with a content delivery network (CDN) to enable requesting end users to subscribe to live streams that have been published to CDN entry points. A reflector is a generalized packet router program. The reflector network preferably comprises a hierarchy of reflectors: at least one reflector located at each entry point to the CDN, at each edge node at which requesting users may be directed by the CDN to obtain live streams, and at various "reflector" nodes located within at least one intermediate layer (in the hierarchy) between the entry points and the edge nodes. The intermediate layer is useful to facilitate delivery of streams for which there is high demand. The edge nodes and each reflector node also include a manager program that arranges for feeds. When an end user is directed to an edge node that is not yet receiving the desired stream, the edge node's manager issues a subscription request to a set of reflector nodes. If the reflector node(s) are already receiving the desired stream, their reflector (s) begin sending it to the requesting edge node. If, however, the reflector node (s) are not already receiving the desired stream, their manager programs issue the subscription request up the hierarchy, ultimately reaching the entry point(s) to start the feed.

Brief Description of Drawings Paragraph:
[0018] FIG. 4 is a flowchart illustrating an operation of the inventive subscription mechanism at the edge node to which a requesting end user has been directed by the CDN;

Brief Description of Drawings Paragraph:
[0019] FIG. 5 is a flowchart illustrating an operation of the subscription mechanism at a set reflector node according to the present invention;

Detail Description Paragraph:
[0025] As described in copending application Ser. No. 09/478,571, which is also incorporated herein by reference, live streaming can be further enhanced by having the CDN send multiple copies of the same stream over different routes from a CDN entry point to the optimal streaming server at the edge of the Internet. These copies are then combined to form one complete, original-quality stream, which is sent from the streaming server to the end users. FIG. 2 illustrates this process in more detail. A broadcast stream 200 is sent to a CDN entry point 202. An entry point, for example, comprises two servers (for redundancy), and each server can handle many streams from multiple content providers. Once the entry point receives the stream, it rebroadcasts copies of the stream to set reflectors 204a-n. The streams are multiplexed and delivered to the set reflectors preferably via UDP (e.g., WMT encapsulated in RTSP encapsulated in UDP over IP). These set reflectors are preferably diverse from a network and geographic standpoint (e.g., at diverse Internet backbone data centers) to ensure fault tolerance. Each set reflector, in turn, rebroadcasts its copy of the stream to each subscribing region, e.g., region 206d, of a set of regions 206a-n. A subscribing region 206d is a CDN region that contains one or more streaming edge nodes 208a-n to which user(s) have been routed by the CDN request-routing mechanism. In other words, set reflectors send their streams to every edge region where they are needed. A CDN region, in this example, includes a set of edge nodes connected by a common backbone 209, e.g., a local area network (LAN). Typically, an edge node, e.g., node 206d, comprises a streaming server 212 and it may include a cache 210. A representative server runs an Intel processor, the Linux operating system and a Real Media or QuickTime Server. For Windows-based platforms, a representative server runs an Intel processor, Windows NT or 2000, and a Windows Media Server. As will be described, the edge node also runs control programs 214 to facilitate the inventive subscription mechanism.

Detail Description Paragraph:
[0026] Each subscribing region, then, simultaneously receives multiple copies of the streamed content. These copies have been sent via separate routes over the Internet, so congestion resulting in dropped packets is unlikely to impact each

copy of the stream equally. As described in copending Ser. No. 09/478,571, each region preferably has a mechanism to recreate in real time an original version of the stream as sent to the entry point. In this way, the technique compensates for the inherently faulty Internet and inherently lossy UDP transport protocol. The reassembly mechanism within each region makes the original, verbatim stream available to every streaming media server within that region. When a user clicks on a CDN-tagged stream, the stream is delivered from the optimal edge node (and, in particular, that node's streaming media server) identified by the CDN's request-routing mechanism. If the CDN maps a user to a node in a region which has not subscribed to that broadcast stream (which, for example, is true for the first connection served from that region), the region automatically notifies the set reflectors and subscribes to that stream using the present invention, as will now be described.

Detail Description Paragraph:
[0028] With reference now to FIG. 3, the main program, called reflector, is a generalized packet moving engine, essentially an application-level router for UDP packets. The reflector moves packets using unicast, multicast or broadcast. Also, the reflector program conditionally sends particular streams to particular places. A reflector can be configured to do this via a static configuration file or, in accordance with the present invention, by learning the configuration via subscription messages on the network. As can been seen in FIG. 3, an entry point 300 includes a reflector program 302a, each set reflector machine 304 includes a reflector program 302b, and each edge node 306 includes a reflector program 302c. The edge node 306 also includes a streaming media server 305 as has been previously described. Entry point 300 also runs a streaming media server.

Detail Description Paragraph:
[0029] Thus, in the preferred embodiment, the reflector program runs in three (3) distinct layers of the streaming network hierarchy comprising entry point(s) at the top level, set reflectors in the intermediate layer, and region(s) of one or more edge nodes at the lower level. Additional set reflector layers can be implemented as well for scalability. Generally, the reflector program works using "listener" and "destination" methods as follows. A listener may be one of these types:

Detail Description Paragraph:
[0046] This is the same as DEMUX, except it consults the subscription list from the local server (the portinfo library) and only sends the packet if it is part of a feed being subscribed to. This allows for higher performance at the edge nodes.

Detail Description Paragraph:
[0065] Thus, in an illustrative embodiment, an entry point runs reflector and portannounce, each set reflector runs reflector, portlisten and submanager, and each edge server platform runs reflector, submanager, and a streaming server that uses the portinfo library. Preferably, a number of reflectors are deployed in strategic places around the Internet to create the entry points. The goal is to have an entry point near the content provider. The content provider sends its live stream to an entry point, which uses portannounce to announce to the set reflectors the availability of the new stream. Because an entry point is just a machine running reflector and does not involve any special hardware, rapid deployment is easy. An entry point preferably comprises two computers with a shared backend, although this is not a requirement. Entry points preferably run a failover mechanism to ensure availability. The set reflectors likewise are located, preferably around the world, in strategic locations. Each set reflector uses its submanager to subscribe to the entry points for feeds as needed, namely, the feeds requested to by the edge nodes. The reflector in each set reflector preferably runs in self-server mode while the submanager therein runs in porttable mode, with the porttable being populated by portlisten. The edge node is what serves actual users. As noted above, the edge node runs a media server (e.g., QTSS, WMS, ~~or~~ Real Media) along with reflector and submanager. Preferably, edge nodes are arranged in

regions, although this is not a requirement. A given region comprises about ten (10) edge machines sharing a common backbone.

Detail Description Paragraph:
[0066] As also illustrated in FIG. 3, a given region 320 may comprise a satellite region that runs a satellite uplink node 315 running reflector 302d and submanager 308c. Generally, satellite uplink nodes do not run servers; rather, they are controlled by the routemaster program and are told to subscribe for the feeds that the routemaster has decided to put on the particular satellite to which the node is connected. Any data received by reflectors in an uplink region preferably is sent to the satellite. Although not meant to be limiting, a given set reflector node is a machine running a Pentium III-class processor, the Linux operating system kernel, and that includes suitable system memory and disk storage to support the application programs described above. A given entry point may have a similar configuration together with additional programs (e.g., an encoder, a content initiator tool, etc.) as needed to publish the live streams to the CDN. A set reflector is made up of set reflector nodes.

Detail Description Paragraph:
[0067] FIG. 4 is a flowchart illustrating the operation of the subscription mechanism at an edge node in the preferred embodiment. The routine begins at step 400 when a user connects to an edge node using, for example, the CDN request-routing mechanism. In particular, the flowchart assumes that the streaming media is ready to be delivered over the streaming CDN, that the requesting user has clicked on a link identifying the stream, and that he or she has been routed to the optimal server by the CDN. The particular methods by which these conditions are achieved are outside the scope of the present invention, and any convenient known or later-developed CDN technology and services may be used for this purpose. At step 402, the streaming media server at the particular edge node to which the user has been routed receives the request and determines which live stream the user wants. At step 404, the streaming media server at the edge node uses the portinfo library to request the stream. A test may then be performed at step 406 to determine whether the streaming media server at the edge node is already receiving the stream (e.g., if another user is playing the stream from that server or a server in the same region). This step may be omitted if multiple subscriptions to a particular stream are treated as a single subscription request. If the outcome of the test is positive, the routine ends. If, however, the outcome of the test at step 406 is negative, the routine continues at step 408.

Detail Description Paragraph:
[0068] At this step, the submanager running on the edge node executes its leader algorithm which, as noted above, involves contacting the submanagers running on other edge nodes in the region and determining a set of leaders. The routine then continues at step 410 to use the DNS subscription method to send subscriptions to the set reflectors. According to the subscription method, when the submanager looks up a given domain name, e.g., n<leader>.r<physicalregion>.ref.akam- ai.com, the DNS server returns the set reflector that is best able to provide the requested megastream and will return one of the three IP addresses based on the n1, n2 or n3 hostname in the domain name. Returning to the flowchart, the routine then continues at step 412 with the set reflectors modifying their subscription lists for the three leaders and then begin sending the new streams requested.

Detail Description Paragraph:
[0072] The subscription mechanism of the present invention may have several variants. Of course, any number of set reflectors may be used within the intermediate layer to provide improved fault tolerance. Moreover, instead of using a static configuration, the mechanism may selected set reflectors dynamically (where a given number of set reflectors are selected from a pool using DNS). Further, entry points may also be selected dynamically instead of merely a hard configuration, i.e., by providing an IP address to a content provider customer. In

addition, it may be desirable to provide intelligent region overflow when a region that is subscribing to many live feeds is unable to receive more streams. A busy region may overflow live traffic to other regions. Moreover, it may be unnecessary to have a submanager at a given edge node to request a given number of incoming streams, especially in well-connected areas. Thus, the submanager may be programmed so that it only requests a stream if the current number of incoming streams is insufficient.

Detail Description Paragraph:
[0075] The low loss characteristics of satellite transport can be exploited by using the techniques described in copending application Ser No. 09/478,571. As was described above, a given CDN region may comprise a satellite region that runs a satellite uplink node running reflector and submanager. The reflector-based satellite system maximizes the usefulness of any particular piece of satellite transit. (The reflector cannot do anything about the cost of satellite, but it can arrange to make the most use out of the expensive bandwidth.) As illustrated in FIG. 7, an uplink center 700 comprises a standard CDN region configuration wherein machines 702a-n share a private backend network 704. These machines run reflector and submanager as has been previously described. Generally, these machines do not run a streaming server and users are not mapped to them. At the downlink, i.e., some other CDN edge region 706, a satellite antenna 708 and its associated hardware 710 is connected to provide its data, preferably in multicast for, onto the backend network 712 in the region. Machines 714a-n are the edge nodes, and each runs reflector and submanager, together with a streaming server as has been described previously.

Detail Description Paragraph:
[0079] Summarizing, in the illustrative embodiment, the reflector network deployment preferably comprises three (3) layers: entry points, set reflectors and edge nodes. Each entry point sends multiple unicasts to the set reflectors; each set reflector receives streams from multiple entry points and then sends a multiplexed stream to a subset of the edge nodes; each edge node is preferably within a CDN region hosting the streaming servers and receives multiple copies of the multiplexed stream from some subset of the set reflectors and then broadcasts them over their backend network so that all servers in the region see all of the streams. Of course, a given region may only include one streaming server (namely, a single edge node), which does not impact the subscription mechanism previously described except to the extent it obviates execution of the leader algorithm. In addition, one of ordinary skill will appreciate that the use of the intermediate layer may be unnecessary with respect to a subscription request for a stream for which there is little demand.

Detail Description Paragraph:
[0080] In addition, the reflector network and its associated subscription mechanism described herein may be used for generalized delivery of any type of data. Thus, for example, the reflector network to publish content provider metadata to CDN edge nodes. As another example, the network and subscription mechanism may be used as a tool for populating edge node caches with content to be served from the CDN.

CLAIMS:

1. A subscription mechanism for use in a content delivery network (CDN) having a set of content provider entry points, an intermediate layer of set reflectors, and a set of edge nodes to which requesting end users are selectively directed to obtain live data streams that are published to the CDN, wherein an edge node includes a server for delivering the live data streams to requesting end users, the mechanism comprising: code operative at the edge node (a) for determining whether the server is already receiving a live data stream being requested by an end user; and (b) for issuing to at least one set reflector a subscription to the live data stream if the server is not already receiving the live data stream; and code

operative at the set reflector for (a) determining whether the set reflector is already receiving the live data stream being requested by the edge node; and (b) for issuing to a given entry point the subscription to the live data stream if the set reflector is not already receiving the live data stream.

2. The subscription mechanism as described in claim 1 wherein the edge node also includes code for routing the live data stream received from at least one set reflector to the server.

3. The subscription mechanism as described in claim 1 wherein the set reflector also includes code for routing the live data stream received from the given entry point to the edge node.

5. The subscription mechanism as described in claim 1 wherein the code operative at the edge node includes code for determining a preferred set of set reflectors to which the subscription is to be issued.

6. A subscription mechanism for use in a content delivery network (CDN) having a set of content provider entry points, an intermediate layer of set reflectors, and a set of edge nodes to which requesting end users are selectively directed to obtain live data streams that are published to the CDN, wherein an edge node includes a server for delivering the live data streams to requesting end users, the mechanism comprising: code operative at the edge node (a) for determining whether the server is already receiving a live data stream being requested by an end user (b) for issuing to at least one set reflector a subscription to the live data stream if the server is not already receiving the live data stream, and (c) for routing the live data stream to the server upon receipt of the live data stream from the set reflector; code operative at the set reflector for (a) determining whether the set reflector is already receiving the live data stream being requested by the edge node, (b) for issuing to a given entry point the subscription to the live data stream if the set reflector is not already receiving the live data stream, and (c) for routing the live data stream to the edge node upon receipt of the live data stream from the given entry point; and code operative at the given entry point for routing the live data stream to the set reflector in response to receipt of the subscription.

7. The subscription mechanism as described in claim 6 wherein the routing code in the edge node, the set reflector and the given entry point is a UDP packet router.

9. The subscription mechanism as described in claim 6 wherein the code operative at the edge node includes code for determining a preferred set of set reflectors to which the subscription is to be issued.

First Hit Fwd Refs

☐ Generate Collection

L17: Entry 1 of 9 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453314 B1
TITLE: System and method for selective incremental deferred constraint processing after bulk loading data

Application Filing Date (1):
19990730

Detailed Description Text (63):
In a parallel multi-node data processing system, a database management system operating in the data processing system can efficiently handle large amounts of data by partitioning the database into a number of database partitions and distributing the database partitioning across the nodes of the data processing system.

First Hit

☐ Generate Collection

L3: Entry 3 of 12 File: PGPB Sep 26, 2002

DOCUMENT-IDENTIFIER: US 20020138640 A1
TITLE: Apparatus and method for improving the delivery of software applications and associated data in web-based systems

Abstract Paragraph:
An improved system for streaming a software application to a plurality of clients comprises a principal server having the software stored thereon as a plurality of blocks and a plurality of intermediate servers between the principal server and the clients. The principal server is configured to stream program and data blocks to downstream devices in accordance with a dynamic prediction of the needs of those devices. The intermediate servers are configured to cache blocks received from connected upstream devices and service requests for blocks issued from downstream devices. In addition, the intermediate servers are further configured to autonomously predict the needs of downstream devices, stream the predicted blocks to the downstream devices, and if the predicted blocks are not present in the intermediate server cache, request those blocks from upstream devices. The intermediate servers can also be configured to make intelligent cache purging decisions with reference to the contents of the caches in other connected devices.

Application Filing Date:
20001222

Summary of Invention Paragraph:
[0004] The Internet, and particularly the world-wide-web, is a rapidly growing network of interconnected computers from which users can access a wide variety of information. Initial widespread use of the Internet was limited to the delivery of static information. A newly developing area of functionality is the delivery and execution of complex software applications via the Internet. There are two basic techniques for software delivery, remote execution and local delivery, e.g., by downloading.

Summary of Invention Paragraph:
[0007] In a local delivery embodiment, the desired application is packaged and downloaded to the user's computer. Preferably, the applications are delivered and installed as appropriate using automated processes. After installation, the application is executed. Various techniques have been employed to improve the delivery of software, particularly in the automated selection of the proper software components to install and initiation of automatic software downloads. In one technique, an application program is broken into parts at natural division points, such as individual data and library files, class definitions, etc., and each component is specially tagged by the program developer to identify the various program components, specify which components are dependent upon each other, and define the various component sets which are needed for different versions of the application.

Summary of Invention Paragraph:
[0008] Once such tagging format is defined in the Open Software Description ("OSD") specification, jointly submitted to the World Wide Web Consortium by Marimba Incorporated and Microsoft Corporation on Aug. 13, 1999. Defined OSD information can be used by various "push" applications or other software distribution

environments, such as Marimba's Castanet product, to automatically trigger downloads of software and ensure that only the needed software components are downloaded to the client in accordance with data describing which software elements a particular version of an application depends on.

Summary of Invention Paragraph:
[0009] Recently, attempts have been made to use streaming technology to deliver software to permit an application to begin executing before it has been completely downloaded. Streaming technology was initially developed to deliver audio and video information in a manner which allowed the information to be output without waiting for the complete data file to download. For example, a full-motion video can be sent from a server to a client as a linear stream of frames instead of a complete video file. As each frame arrives at the client, it can be displayed to create a real-time full-motion video display. However, unlike the linear sequences of data presented in audio and video, the components of a software application may be executed in sequences which vary according to user input and other factors.

Summary of Invention Paragraph:
[0012] One challenge in implementing a predictive streaming system is maintaining an acceptable rate of data delivery to a client, even when many clients are executing streaming applications. A technique which has been used to improve the delivery time of Internet hosted data accessed by many users is to use caching techniques. In standard Internet-based web-page distribution systems, caching systems are linked between a primary server hosting the web site and the end users or clients, with each cache server servicing a number of corresponding clients. These cache servers are used to store web pages that have been requested by a client from a principal server. Each time a client requests a particular web page, the request is processed by the respective cache server which is servicing the client. If the requested page is present in the cache server, the page is extracted from the cache and returned to the client. If the requested page has not been previously accessed by any of the clients corresponding to the particular cache server, the cache server forwards the request to the primary server to download the page from the Web site, stores the retrieved web page, and serves that page to the client.

Summary of Invention Paragraph:
[0015] The present invention relates generally to a method and system for improving the delivery of software applications and associated data, which can be stored in databases, via a network, such as the Internet. One or more intermediate tiers of intelligent caching servers are placed between a principal application server and the streaming application clients. The intermediate servers store streamed blocks, such as software application modules or streamlets and other database modules, as they are transmitted from the principal server to a client. As a result, further requests by the same client or other clients associated with the intermediate servers for previously stored information can be streamed from the intermediate servers without accessing the principal server.

Detail Description Paragraph:
[0060] As discussed above, various techniques can be used to predict the order in which a client will require various program elements during execution. The following is a more detailed discussion of a particular technique for predicting this order for use in determining an order in which program elements should be streamed to a particular client. As discussed above, this information can be used in the presently disclosed system, along with additional information, such as the contents of related intermediate servers, download times, etc., by each intermediate server to determine the most appropriate streaming sequences to downstream devices and to further determine program elements which should be requested from upstream devices in anticipation of future needs.

Detail Description Paragraph:

[0062] To minimize module download delays experienced by a user, module "E" may be transparently streamed from a server to the client computer before it is required at the client. Transparent streaming allows future module use to be predicted and modules to be downloaded while other interrelated modules "A" are executing. Referring to FIG. 6, the execution order of application modules "A" through "H" can be visualized as a directed graph 600 rather than a linear sequence of modules. For example, as illustrated by the graph, after module "A" is executed, execution can continue at module "B," "D," or "E." After module "B" is executed, execution can continue at module "C" or "G." The execution path may subsequently flow to additional modules and may return to earlier executed modules.

First Hit

Search Forms

Search Results

Help

User Searches

Preferences

Logout

Generate Collection

Result 4 of 26

File: PGPB

Nov 21, 2002

DOCUMENT-IDENTIFIER: US 20020174207 A1

TITLE: Self-healing hierarchical network management system, and methods and apparatus therefor

Application Filing Date:
20010228

Detail Description Paragraph:
[0115] In summary, the Node Manager provides NMS interface and local node management, as well as providing signaling, routing and fault protection functions (all using the Node Manager's application software), provides real-time LCM provisioning, receives monitored parameters and alarms/faults from each LCM, aggregates monitored parameters and alarms/faults from each line card into a node-wide view, processes node-to-node communication messages, provides remote software download capability, distributes new software to all LCMs, is expandable to utilize a more powerful CPU (through plug-in processor 512), such as of RISC design, is built on a Real-Time Operating System (RTOS), provides intra-OTS networking support (e.g., LAN connectivity to LCMs), and provides node-to-node networking support.

First Hit

☐ Generate Collection

L5: Entry 3 of 7 File: PGPB Feb 6, 2003

DOCUMENT-IDENTIFIER: US 20030026254 A1
TITLE: Method and apparatus for large payload distribution in a network

Abstract Paragraph:
Large payload files are selectively partitioned in blocks and the blocks distributed to a plurality of distribution stations at the edge of the network qualified to have the data. Each qualified station decides how much and what portion of the content to save locally, based on information such as network location and environment, usage, popularity, and other distribution criteria defined by the content provider. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server, a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, a cluster of distribution servers at the distribution station may download the non-resident portions of the file as the application server is servicing the user. The download may be in parallel and usually from the least congested nodes. New nodes added to the network learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage devices from the nearest and least congested nodes without interrupting network operation. Each node manages its local storage and decides what content to prune based on information such as usage patterns.

Cross Reference to Related Applications Paragraph:
[0001] This application claims the benefit of U.S. Provisional Application No. 60/266,286, filed on Oct. 26, 2000, entitled "Large Payload Delivery Networks Having Integrated Content Management Services," the specification of which is herein incorporated by reference.

Summary of Invention Paragraph:
[0005] Network-based content delivery that relies on a single source to simultaneously distribute various types of information to multiple remote locations may, depending on the size of files being transferred, encounter network-loading problems around the server or the server itself may be over tasked. For example, since transferring a small file (e.g., a web-page) usually takes only a few seconds, the massive distribution of a small file from one source to thousands of destination locations may not create large impact on the network traffic near the source. Transferring a large file (i.e., a large payload), in contrast, can take tens of minutes to hours. If the distribution of such payloads relies on a single source, the network performance near the source, and the subsequent delivery of content, could degrade severely and become unacceptable.

Summary of Invention Paragraph:
[0007] The fast-paced expansion of the broadband industry has fueled the push for rich media (e.g., full length movies, video, or other types of multimedia data). Broadband technology brings high-speed connection capabilities for content delivery to remote users hence large payloads can be transferred faster. Also, broadband technology makes it possible to send audio and/or video data using streaming media whereby the data is sent in streams for real-time playback, for example. Thus, the

quality of rich media at the user's terminal, more than that of any other type of information, is now more dependent on the performance capabilities of the delivery technology. In order to minimize delivery delays, network congestion, and other related problems, some systems attempt to locate content on server systems that are located in close proximity to, i.e., a few hubs of connections away from the end-users. These server locations approximately define the concept known as the "edge" of the network. For example, the Internet service providers are in close proximity to the end-user thus may be regarded as being at the edge of the network. When servers are placed in such locations, the servers are said to be at the edge of the network. End-user systems that are configured to obtain content from network nodes located at the edge of the network are therefore beyond the edge of the network (a.k.a. last mile). However, it is important to note that systems located beyond the edge of the network are still coupled to the network and capable of communicating with the server computers located at the edge. Placing content at the edge of the network is advantageous because it can reduce the latency in servicing users located beyond the edge. Current approaches for delivering large payloads to the "edge" consist of mirroring or caching. These approaches and the limitations inherent in each approach will now be discussed in detail so as to give the reader an understanding of the advancements made by the invention.

Summary of Invention Paragraph:
[0035] Therefore, there is a need to address the cost, scalability, and load-balancing issues associated with large payload delivery to the edge of the network. However, before discussing the present invention, a general overview of how files are handled in different operating systems is presented.

Summary of Invention Paragraph:
[0041] An embodiment of the invention provides an improved mechanism for distributing large files throughout a computer network and delivering such files to an end-user system. When the invention is implemented it provides multiple users with a way to obtain access to large payload files without overburdening network resources. If, for example, a user wishes to download a large file such as a video file an embodiment of the invention provides a way to deliver that video file to the requesting user without putting a strain on the network. The system accomplishes this by breaking the large file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of large data files. Thus, the portions of large data file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations are referred to by those of ordinary skill in the art as the edge of the network.

Summary of Invention Paragraph:
[0043] In one embodiment of the invention, the system is optimized so that large payload files can be distributed across existing networks (including the Internet and corporate intranets) using a transport layer network overlay to push content to the edge of the network. Specifically, the embodiments of the invention improve large payload delivery performance, scalability, reliability, and availability.

Summary of Invention Paragraph:
[0044] As mentioned above, one embodiment of the invention breaks the large payload files into multiple portions. This may be accomplished by selectively partitioning the large payload file into blocks that are replicated and distributed to a plurality of distribution stations (a.k.a. nodes) at the edge of the network. Each distribution station is configured to determine how much of the content to save locally, based on information such as usage, popularity, etc. The content provider defines what distribution stations are qualified to function as distribution stations and may also define other distribution criteria. Distribution stations in the network manage storage and transfer content (e.g., portions of large payload

files) and other information to one another. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server (e.g., a streaming server), a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, the distribution stations may download the non-resident portions of the file as the application server is servicing the user. The download of the non-resident blocks may be in parallel and usually from the least congested nodes. The entire process is transparent to the user.

Summary of Invention Paragraph:
[0047] In one or more embodiments, new nodes may be added to the network without service interruption. As the new nodes are added, they learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage from the nearest and least congested nodes. Thus, a node could be added to the network and it would be up and running after self-initialization.

Summary of Invention Paragraph:
[0048] In one or more embodiments, the portions and amount of a large payload file maintained at each node depends on the available storage, popularity of the content, distribution criteria by the content provider, etc. Thus, least likely to be used blocks of a large payload file may be pruned (i.e., deleted from local storage) to make room for other highly desirable content. However, although the least likely to be used blocks of a file are pruned, the entire content of a large payload file may be maintained at a node in the scalable content delivery network, so long as the content provider wants the content to remain in the network.

Brief Description of Drawings Paragraph:
[0053] FIG. 5 is an illustration of a scalable content delivery network for delivering large payloads according to an embodiment of the present invention.

Brief Description of Drawings Paragraph:
[0061] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within the network of the present invention.

Detail Description Paragraph:
[0075] When the invention is implemented in accordance with one embodiment of the invention it provides end-user systems with a way to access large payload files without overburdening the network utilized by the end-user system to transmit data. In one embodiment of the invention, the system accomplishes this by breaking the large payload file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of the large payload files. Thus, portions of the large payload file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations minimize the latency associated with delivering the file to the end-user system and are referred to herein as the edge of the network.

Detail Description Paragraph:
[0077] FIG. 5 provides a view of a scalable content delivery network (SCDN) for delivering large payloads according to an embodiment of the present invention. SCDN 500 may be a network such as the Internet which conceptually includes a network core 505 (i.e., the backbone), intermediate network segments 510 ranging "near" and "far" from the core, and network segments "far" from core 520-A through 520-C (collectively 520). "Near" and "far" relate to distance and are intended to indicate relative path latencies (short or long, respectively) to the core, such latencies generally depend on the number of intermediate hubs (e.g., switches,

routers, and the like) that are traversed to reach the high-speed backbones that form the core of the network and through which much of the network traffic is routed. Note that each intermediate hub may perform some limited processing, which adds latency, before forwarding the traffic to the next hub.

Detail Description Paragraph:
[0078] FIG. 5 shows a plurality of Content Provider Client (CPC) systems 530, a plurality of End-User Client (EUC) systems 550, and one or more Content Management Servers (CMS) 570, all located beyond Network Edge 501. In general, the content provider client 530 may be connected (or assigned) to a content management server 570, which in turn is connected to its assigned distribution center 540. A content provider uploads and/or manages large payload files in the SCDN 500 through its CPC 530. The EUC 550 provides the end-user access to files in SCDN 500. For example, EUC 550 may be a browser running on the end-user's local computer.

Detail Description Paragraph:
[0079] Network Edge 501 generally may be far from network core 505. However, the distance (i.e., path latency) between the core and the edge may not be uniform and may vary considerably for a given CPC or EUC. One embodiment of the present invention places a plurality of Distribution Centers (DC) 540A-540I for maintaining large payloads at the edge of the network thereby resolving the latency issue. Large payload content from a content provider is pushed from one distribution center to other distribution centers at the edge of the network. An end-user seeking access to a large payload is serviced (via an application server) from the nearest distribution center containing the desired content. By distributing content to the end-user (e.g., at EUC 550) via a plurality of Application Servers 560 and distribution centers 540 at the edge, path latency is minimized. Thus, large payload distribution involves obtaining a large payload file from a content provider and geographically placing such file at the distribution centers which are at or as close to the edge of the network as possible.

Detail Description Paragraph:
[0082] Content management server 570 may be connected to any node on the tree. Thus, although a content management server and a distribution center may not be collocated, the content management server gives the content provider a vehicle to upload large files (e.g., video) to the distribution centers. In one embodiment, the content management server is a computer that processes the content provider's large payload file for distribution in the network. In another embodiment, the content management server may, for example, be a subset of tools (e.g., machine independent objects) that allows upload of content to the network; thus, the tools may be shipped from a server to the content providers client's computer for processing and distribution of the large payload file in the network. After a content provider loads the large payload file into the content management server, the CMS may process the file and forward it to the distribution center.

Detail Description Paragraph:
[0085] The Virtual File Control System (VFCS) 702 is able to piece the original (large payload) file back together from the block files. As will be explained later, all the blocks of the large payload file need not be stored at one distribution center, however, the entire file is available within the SCDN. When an end user connects to application server 721 (e.g., a streaming server), the VFCS creates a virtual appearance that the entire file is available at that node. For example, assuming only fifteen percent of a two-gigabyte file is stored in storage 711-713, the VFCS makes streaming server 721 think that the entire two gigabytes is available at the location. Thus, streaming server 721 may start playing the file. As the file is being played, VFCS communicates with DS to locate and retrieve the remaining portions of the file from other nodes in the network.

Detail Description Paragraph:
[0110] To distribute a file, a content provider sets specific distribution criteria

for that file. After the distribution server (DS) stores the uploaded large payload file as blocks, the content provider requests, through the content management server, that the DS distribute the file to other nodes in the SCDN, i.e., to push the content to the edge of the network. The distribution is in accordance with specific distribution criteria set by the content provider and may use the file distribution protocol (FDP) previously described. The distribution criteria may specify regions (e.g., Europe), specific nodes, and other information as desired by the content provider to control distribution of the content. For example, the distribution criteria may include information found in the nodes attribute or rolled up attribute bitmap.

Detail Description Paragraph:
[0111] The file distribution proceeds as follows: (1) The DS responds to the content provider's request to distribute a large payload file by sending a notification (i.e., a distribution request) to its neighbors to announce the existence and the distribution criteria of the file; (2) "Qualified" neighbors (i.e., those that meet the criteria) download several portions of the file during this initial distribution process; (3) The notification is then passed on from neighbor to neighbor, but not back to the neighbor from which the distribution request is received; (4) Each neighbor performs steps 2 and 3 until it encounters a leaf node or a "terminating" node. Thus, the distribution of the file in the network is done in stages.

Detail Description Paragraph:
[0113] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within an SCDN. A content provider uploads a large payload file into the content management server (CMS) 570, which is connected to node B of the SCDN, using any content publishing and management software running on the content provider's client system (CPC) 530. The content provider also uploads the distribution criteria onto CMS 570. Content management server 570, as previously described, divides the uploaded file into track files and issues a command similar to the FDP "put" command for each track file to the distribution server located in node B. In other embodiments, the CMS may be connected to any node of the SCDN. At node B, the DS divides the track files into block files for local storage. The full copy of the file is shown at Node Bas a filled in dot. The CMS then issues an FDP command of the type "distribute" to the distribution server at node B. In response to the distribute command, the DS issues a command to its neighboring nodes A , D, and E to replicate the content (e.g., using the "replicate" command of the FDP). Node D examines the replicate packet and decides its not supposed to have the content thus it passes the replicate command to its neighbor, node H. Nodes A , E, and H examine the replicate packet and decide they all match the distribution criteria (i.e., they are "qualified" nodes). When ready, nodes A , E, and H issue commands to retrieve a portion of the file from the nearest node (e.g., node B) in the SCDN. Nodes E and H are leaf nodes thus they do not propagate the replicate command. However, node A is the root node with child nodes Band C. Node A may not send the replicate command back to node B, because it is the originating node. However, node A may send the replicate request to node C. Node C checks the distribution criteria and decides it's a qualified node therefore it retrieves a portion of the file from the nearest nodes (e.g., the nearest of nodes A, B, E, and H) containing the needed data. Node C subsequently sends the replicate command to nodes F and C. Node F is qualified thus it retrieves a portion of the file from the nearest nodes having the data (e.g. nodes B or C). Nodes G and/are not qualified thus they receive nothing. Node G is a terminating node because the rolled-up attribute of its branch does not satisfy the distribution criteria. This initial replication process continues until all the qualified nodes in SCDN are at least partially populated. In one or more embodiments, the same portion (e.g., blocks) of the large payload file is contained in at least one node of the SCDN. Preferably, a plurality of nodes maintains the same portion thereby creating redundancy and preventing loss of any portion of the large payload file when one or more nodes or storage volumes become unavailable. For example, when a storage volume (or device)

becomes unavailable (i.e., lost), a DS at that station need not take any special action to recover contents of the damaged volume since the portions of large payload files stored and hence lost in that volume are automatically downloaded from other network nodes upon demand to service a user request. The distribution servers also relay control information of a failed station to neighbors of the failed station to prevent improper termination of control commands.

Detail Description Paragraph:
[0115] The FDP uses the content provider's distribution criteria to direct the distribution of the large payload file in whole or in part to all nodes in the network meeting the provider's distribution criteria. A distribution request can start from any node in the tree, and traverses up and down the tree until it reaches a leaf node or arrives at a terminating node. For any node having the appropriate attributes, the file is partially downloaded from the nearest neighbors that meet specific performance criteria if those neighbors contain the portion of the file to be downloaded. The nearest neighbor when downloading content is not necessarily the nearest in the virtual tree but nearest in terms of distance. This prevents massive transfers from the node at which the file is initially uploaded. Moreover, the staging nature of the distribution prevents excessive demands on the network around the initial node (e.g., node B). By delivering smaller blocks and only a partial file this delivery method reduces network load. Additionally, because the distribution requests stop progressing through the SCDN when they arrive at a "terminating" node, the present invention prevents unnecessary distribution request packets from flooding the network.

Detail Description Paragraph:
[0132] A SCDN in accordance with an embodiment of the present invention is highly scalable. For example, when a new node is added to the SCDN, it downloads the initial content it needs by employing one of several different adaptable initialization processes. In one embodiment of the invention, an "Auto-Initialization" process is used. When a node is added to an SCDN, it is given a set of attributes. In the auto-initialization process, as soon as the node is connected to the network, it issues an FDP "Learn" or similar request to all its neighbors. The node encodes its attributes in the learn request. The neighbors offer content, consistent with the new node's attributes, to it for downloading. The neighbors then pass on the new nodes learn request to all of their neighbors, which take similar action. Thus, the new node's learn request traverses the entire network and all the nodes in the network respond to the learn request if they have contents appropriate for the new node. The new node collects all the information, downloads the necessary initial contents, and is now a functioning element of the SCDN.

Detail Description Paragraph:
[0139] As in the earlier embodiments, the stations of SCDN 1400 are organized in a logical virtual tree structure in which each node in the tree has a set of attributes. Thus, each Station has an attribute set that is stored in the node and can be represented in any convenient data structure, e.g., the attribute set can be represented as an attribute bitmap. Furthermore, each Station (i.e., node) also contains a representation of the rolled up attribute set of each of the station's child-Stations. This representation is called the "Rolled Up Set of Attributes" and any convenient data structure can be used for it, e.g., a "Rolled Up Bitmap", which may be the defined as the "binary OR" combination of all rolled up attribute bitmaps from the child-Stations. The distribution servers within a Distribution Server Cluster use the attribute bitmap to distribute and route portions of large payload files and they use the aggregated rolled-up attribute bitmap to terminate unnecessary propagation of messages. One of the Stations in an SCDN is designated the "Central Station". The Central Station holds an attribute database table that matches text strings to bit positions, e.g., a reference table. Central Station 1420 is not necessarily a data repository for all content but may contain some or all the content.

Detail Description Paragraph:
[0209] There are multiple storage volumes in the shared Storage System 1530 (FIG. 15). These volumes contain a number of important databases illustrated in FIG. 20 such as: The Content Provider Data Table, File Distribution Criteria Table, the File Metadata Database, and the content block files. These databases are stored independently and may be mapped arbitrarily anywhere within shared Storage System 1530. The Content Provider Data Table and File Distribution Criteria Table also exist in each Content Management Server system. The tables in the Content Management Server only include the data of the content providers that are assigned to that CMS. The Content Provider Data Table at each station includes the information of all the content providers of the SCDN, and the File Distribution Criteria Table includes only the media files that are replicated to the station.

Detail Description Paragraph:
[0218] Distribution servers communicate with one another and the Content Management application in order to transfer large payload files in the SCDN. The DSs modify entries in the File Metadata Database when they add or remove blocks in the shared Storage System. A sequence server, which serves essentially as a lock manager, may be used to synchronize access to the file metadata database by multiple DSs, VFCS servers, storage managers, etc., to prevent possible race (e.g., conflict) condition. VFCS servers use information in the File Metadata Database to assemble and multiplex appropriate blocks into files for the Application Servers. The Storage Management Subsystem watches the available shared storage, the content provider's reserved storage, and the usage logs. It initiates the removal of less popular content to make room for more popular and new content when available storage is running low. It does this by instructing the DS to remove some of their associated blocks that are least likely to be used. It accesses the File Metadata Database to determine how many and which blocks it will request to be deleted.

Detail Description Paragraph:
[0267] The Content Management Server is a tool that acts as a bridge between the Content Management Client tool and the distribution server. It communicates with a DS in its assigned station using the FDP protocol (e.g., using the "put", "distribute", and "clean" commands). When the Content Management Client tool uploads a new file, the Content Management Server tool distributes (i.e., injects) the file into the SCDN via its assigned DS using FDP "put" and "distribute" packets. Content Management Server saves content provider information, and content distribution criteria in its database. While content is distributed to the SCDN, the DSs involved in processing the distribution request store information related to the files and their constituent portions in the File Distribution Criteria Database and the File Metadata Database. The files and their constituent blocks are stored in the content repository distributed in the SCDN, also by DS involved in processing the request. The Content Management Server uses the FDP "clean" packet to remove a file from an SCDN. When a DS in the SCDN receives a "clean" packet, it removes the relevant information from File Distribution Criteria Database, File Metadata Database, and the actual content blocks from the content repository.

CLAIMS:

1. A method for distributing content to a plurality of network nodes comprising: obtaining a large payload file having content from a client for distribution in a network having a plurality of nodes, said client providing distribution criteria for said content; dividing said content into a plurality of block files; distributing a subset of said plurality of block files to a plurality of qualified nodes, wherein each of said qualified nodes conforms to said distribution criteria set by said client.

15. A method for distributing content to a plurality of network nodes comprising: obtaining a large payload file having content from a content provider for distribution in a network having a plurality of nodes; said content provider

providing distribution criteria for said content, said distribution criteria having information about which of said plurality of nodes are authorized to have said content; dividing said content into a plurality of block files each having a block size that maximizes playback of said content; storing said plurality of block files in a plurality of local storage devices such that said plurality of block files are distributed over said plurality of local storage devices to load balance said plurality of local storage devices during input/output operations; distributing a subset of said plurality of block files from said local storage devices to each one of said plurality of nodes qualified to receive said file content, wherein said nodes qualified are those nodes that conform to said distribution criteria set by said content provider.

16. A method for distributing content to a plurality of network nodes comprising: obtaining a large payload file comprising file content from a client for distribution in a network having a plurality of nodes, said client providing distribution criteria for said file content, said distribution criteria having information about which of said plurality of nodes are authorized to have said content; dividing said file content into a plurality of block files having a block size that maximizes playback of said content; storing said plurality of block files in a plurality of local storage devices such that said plurality of block files are distributed over said plurality of local storage devices to load balance said plurality of local storage devices during input/output operations; distributing a subset of said plurality of block files from said local storage devices to each one of said plurality of nodes qualified to receive said file content, wherein said nodes qualified are those nodes that conform to said distribution criteria set by said client, wherein each qualified node receiving said subset of said plurality of block files distributes said subset into a plurality of local storage devices of said receiving node.

17. A computer program product comprising: a computer usable medium comprising computer readable code for distributing content to a plurality of network nodes, said computer readable program code configured to: obtain a large payload file having content from a client for distribution in a network having a plurality of nodes, said client providing distribution criteria for said content; divide said content into a plurality of block files; distribute a subset of said plurality of block files to a plurality of qualified nodes, wherein each of said qualified nodes conforms to said distribution criteria set by said client.

31. An apparatus for distributing content to a plurality of network nodes comprising: a network having a plurality of nodes, each of said plurality of nodes having a plurality of components capable of communicating with one another, said plurality of components comprising one or more first servers in a first server cluster; a third server obtaining a large payload file having content from a client for distribution in said network, said client providing distribution criteria for said content, said third server transmitting said large payload file to a first server of said one or more first servers in said first server cluster, said first server dividing said content into a plurality of block files and distributing a subset of said plurality of block files to a plurality of qualified nodes, wherein each of said qualified nodes conforms to said distribution criteria set by said client.

First Hit

☐ Generate Collection

L5: Entry 2 of 7 File: PGPB Feb 13, 2003

DOCUMENT-IDENTIFIER: US 20030031176 A1
TITLE: Method and apparatus for distributing large payload file to a plurality of storage devices in a network

Abstract Paragraph:
Large payload files are selectively partitioned in blocks and the blocks distributed to a plurality of distribution stations at the edge of the network qualified to have the data. Each qualified station decides how much and what portion of the content to save locally, based on information such as network location and environment, usage, popularity, and other distribution criteria defined by the content provider. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server, a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, a cluster of distribution servers at the distribution station may download the non-resident portions of the file as the application server is servicing the user. The download may be in parallel and usually from the least congested nodes. New nodes added to the network learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage devices from the nearest and least congested nodes without interrupting network operation. Each node manages its local storage and decides what content to prune based on information such as usage patterns.

Cross Reference to Related Applications Paragraph:
[0001] This application is a divisional of U.S. application Ser. No. 09/681,644, filed on May 15, 2001, entitled "Method and Apparatus For Large Payload Distribution in a Network," which claims the benefit of U.S. Provisional Application No. 60/266,286, filed on Oct. 26, 2000, entitled "Large Payload Delivery Networks Having Integrated Content Management Services," the specification of which is herein incorporated by reference.

Summary of Invention Paragraph:
[0005] Network-based content delivery that relies on a single source to simultaneously distribute various types of information to multiple remote locations may, depending on the size of files being transferred, encounter network-loading problems around the server or the server itself may be over tasked. For example, since transferring a small file (e.g., a web-page) usually takes only a few seconds, the massive distribution of a small file from one source to thousands of destination locations may not create large impact on the network traffic near the source. Transferring a large file (i.e., a large payload), in contrast, can take tens of minutes to hours. If the distribution of such payloads relies on a single source, the network performance near the source, and the subsequent delivery of content, could degrade severely and become unacceptable.

Summary of Invention Paragraph:
[0007] The fast-paced expansion of the broadband industry has fueled the push for rich media (e.g., full length movies, video, or other types of multimedia data). Broadband technology brings high-speed connection capabilities for content delivery

to remote users hence large payloads can be transferred faster. Also, broadband technology makes it possible to send audio and/or video data using streaming media whereby the data is sent in streams for real-time playback, for example. Thus, the quality of rich media at the user's terminal, more than that of any other type of information, is now more dependent on the performance capabilities of the delivery technology. In order to minimize delivery delays, network congestion, and other related problems, some systems attempt to locate content on server systems that are located in close proximity to, i.e., a few hubs of connections away from the end-users. These server locations approximately define the concept known as the "edge" of the network. For example, the Internet service providers are in close proximity to the end-user thus may be regarded as being at the edge of the network. When servers are placed in such locations, the servers are said to be at the edge of the network. End-user systems that are configured to obtain content from network nodes located at the edge of the network are therefore beyond the edge of the network (a.k.a. last mile). However, it is important to note that systems located beyond the edge of the network are still coupled to the network and capable of communicating with the server computers located at the edge. Placing content at the edge of the network is advantageous because it can reduce the latency in servicing users located beyond the edge. Current approaches for delivering large payloads to the "edge" consist of mirroring or caching. These approaches and the limitations inherent in each approach will now be discussed in detail so as to give the reader an understanding of the advancements made by the invention.

Summary of Invention Paragraph:
[0038] Therefore, there is a need to address the cost, scalability, and load-balancing issues associated with large payload delivery to the edge of the network. However, before discussing the present invention, a general overview of how files are handled in different operating systems is presented.

Summary of Invention Paragraph:
[0044] An embodiment of the invention provides an improved mechanism for distributing large files throughout a computer network and delivering such files to an end-user system. When the invention is implemented it provides multiple users with a way to obtain access to large payload files without overburdening network resources. If, for example, a user wishes to download a large file such as a video file an embodiment of the invention provides a way to deliver that video file to the requesting user without putting a strain on the network. The system accomplishes this by breaking the large file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of large data files. Thus, the portions of large data file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations are referred to by those of ordinary skill in the art as the edge of the network.

Summary of Invention Paragraph:
[0046] In one embodiment of the invention, the system is optimized so that large payload files can be distributed across existing networks (including the Internet and corporate intranets) using a transport layer network overlay to push content to the edge of the network. Specifically, the embodiments of the invention improve large payload delivery performance, scalability, reliability, and availability.

Summary of Invention Paragraph:
[0047] As mentioned above, one embodiment of the invention breaks the large payload files into multiple portions. This may be accomplished by selectively partitioning the large payload file into blocks that are replicated and distributed to a plurality of distribution stations (a.k.a. nodes) at the edge of the network. Each distribution station is configured to determine how much of the content to save locally, based on information such as usage, popularity, etc. The content provider

defines what distribution stations are qualified to function as distribution stations and may also define other distribution criteria. Distribution stations in the network manage storage and transfer content (e.g., portions of large payload files) and other information to one another. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server (e.g., a streaming server), a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, the distribution stations may download the non-resident portions of the file as the application server is servicing the user. The download of the non-resident blocks may be in parallel and usually from the least congested nodes. The entire process is transparent to the user.

Summary of Invention Paragraph:
[0051] Thus, a node could be added to the network and it would be up and running after self-initialization.

Summary of Invention Paragraph:
[0052] In one or more embodiments, the portions and amount of a large payload file maintained at each node depends on the available storage, popularity of the content, distribution criteria by the content provider, etc. Thus, least likely to be used blocks of a large payload file may be pruned (i.e., deleted from local storage) to make room for other highly desirable content. However, although the least likely to be used blocks of a file are pruned, the entire content of a large payload file may be maintained at a node in the scalable content delivery network, so long as the content provider wants the content to remain in the network.

Brief Description of Drawings Paragraph:
[0057] FIG. 5 is an illustration of a scalable content delivery network for delivering large payloads according to an embodiment of the present invention.

Brief Description of Drawings Paragraph:
[0065] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within the network of the present invention.

Detail Description Paragraph:
[0079] When the invention is implemented in accordance with one embodiment of the invention it provides end-user systems with a way to access large payload files without overburdening the network utilized by the end-user system to transmit data. In one embodiment of the invention, the system accomplishes this by breaking the large payload file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of the large payload files. Thus, portions of the large payload file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations minimize the latency associated with delivering the file to the end-user system and are referred to herein as the edge of the network.

Detail Description Paragraph:
[0081] FIG. 5 provides a view of a scalable content delivery network (SCDN) for delivering large payloads according to an embodiment of the present invention. SCDN 500 may be a network such as the Internet which conceptually includes a network core 505 (i.e., the backbone), intermediate network segments 510 ranging "near" and "far" from the core, and network segments "far" from core 520-A through 520-C (collectively 520). "Near" and "far" relate to distance and are intended to indicate relative path latencies (short or long, respectively) to the core, such latencies generally depend on the number of intermediate hubs (e.g., switches, routers, and the like) that are traversed to reach the high-speed backbones that

form the core of the network and through which much of the network traffic is routed. Note that each intermediate hub may perform some limited processing, which adds latency, before forwarding the traffic to the next hub.

Detail Description Paragraph:
[0082] FIG. 5 shows a plurality of Content Provider Client (CPC) systems 530, a plurality of End-User Client (EUC) systems 550, and one or more Content Management Servers (CMS) 570, all located beyond Network Edge 501. In general, the content provider client 530 may be connected (or assigned) to a content management server 570, which in turn is connected to its assigned distribution center 540. A content provider uploads and/or manages large payload files in the SCDN 500 through its CPC 530.

Detail Description Paragraph:
[0084] Network Edge 501 generally may be far from network core 505. However, the distance (i.e., path latency) between the core and the edge may not be uniform and may vary considerably for a given CPC or EUC. One embodiment of the present invention places a plurality of Distribution Centers (DC) 540A-5401 for maintaining large payloads at the edge of the network thereby resolving the latency issue. Large payload content from a content provider is pushed from one distribution center to other distribution centers at the edge of the network. An end-user seeking access to a large payload is serviced (via an application server) from the nearest distribution center containing the desired content. By distributing content to the end-user (e.g., at EUC 550) via a plurality of Application Servers 560 and distribution centers 540 at the edge, path latency is minimized. Thus, large payload distribution involves obtaining a large payload file from a content provider and geographically placing such file at the distribution centers which are at or as close to the edge of the network as possible.

Detail Description Paragraph:
[0087] Content management server 570 may be connected to any node on the tree. Thus, although a content management server and a distribution center may not be collocated, the content management server gives the content provider a vehicle to upload large files (e.g., video) to the distribution centers. In one embodiment, the content management server is a computer that processes the content provider's large payload file for distribution in the network. In another embodiment, the content management server may, for example, be a subset of tools (e.g., machine independent objects) that allows upload of content to the network; thus, the tools may be shipped from a server to the content providers client's computer for processing and distribution of the large payload file in the network. After a content provider loads the large payload file into the content management server, the CMS may process the file and forward it to the distribution center.

Detail Description Paragraph:
[0090] The Virtual File Control System (VFCS) 702 is able to piece the original (large payload) file back together from the block files. As will be explained later, all the blocks of the large payload file need not be stored at one distribution center, however, the entire file is available within the SCDN. When an end user connects to application server 721 (e.g., a streaming server), the VFCS creates a virtual appearance that the entire file is available at that node. For example, assuming only fifteen percent of a two-gigabyte file is stored in storage 711-713, the VFCS makes streaming server 721 think that the entire two gigabytes is available at the location. Thus, streaming server 721 may start playing the file. As the file is being played, VFCS communicates with DS to locate and retrieve the remaining portions of the file from other nodes in the network.

Detail Description Paragraph:
[0116] To distribute a file, a content provider sets specific distribution criteria for that file. After the distribution server (DS) stores the uploaded large payload file as blocks, the content provider requests, through the content management

server, that the DS distribute the file to other nodes in the SCDN, i.e., to push the content to the edge of the network. The distribution is in accordance with specific distribution criteria set by the content provider and may use the file distribution protocol (FDP) previously described. The distribution criteria may specify regions (e.g., Europe), specific nodes, and other information as desired by the content provider to control distribution of the content. For example, the distribution criteria may include information found in the nodes attribute or rolled up attribute bitmap.

Detail Description Paragraph:
[0117] The file distribution proceeds as follows: (1) The DS responds to the content provider's request to distribute a large payload file by sending a notification (i.e., a distribution request) to its neighbors to announce the existence and the distribution criteria of the file; (2) "Qualified" neighbors (i.e., those that meet the criteria) download several portions of the file during this initial distribution process; (3) The notification is then passed on from neighbor to neighbor, but not back to the neighbor from which the distribution request is received; (4) Each neighbor performs steps 2 and 3 until it encounters a leaf node or a "terminating" node. Thus, the distribution of the file in the network is done in stages.

Detail Description Paragraph:
[0119] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within an SCDN. A content provider uploads a large payload file into the content management server (CMS) 570, which is connected to node B of the SCDN, using any content publishing and management software running on the content provider's client system (CPC) 530. The content provider also uploads the distribution criteria onto CMS 570. Content management server 570, as previously described, divides the uploaded file into track files and issues a command similar to the FDP "put" command for each track file to the distribution server located in node B. In other embodiments, the CMS may be connected to any node of the SCDN. At node B, the DS divides the track files into block files for local storage. The full copy of the file is shown at Node B as a filled in dot. The CMS then issues an FDP command of the type "distribute" to the distribution server at node B. In response to the distribute command, the DS issues a command to its neighboring nodes A , D, and Eto replicate the content (e.g., using the "replicate" command of the FDP). Node Dexamines the replicate packet and decides its not supposed to have the content thus it passes the replicate command to its neighbor, node H. Nodes A , E, and Hexamine the replicate packet and decide they all match the distribution criteria (i.e., they are "qualified" nodes). When ready, nodes A, E, and H issue commands to retrieve a portion of the file from the nearest node (e.g., node B) in the SCDN. Nodes E and Hare leaf nodes thus they do not propagate the replicate command. However, node A is the root node with child nodes Band C. Node A may not send the replicate command back to node B, because it is the originating node. However, node A may send the replicate request to node C. Node C checks the distribution criteria and decides it's a qualified node therefore it retrieves a portion of the file from the nearest nodes (e.g., the nearest of nodes A, B, E, and H) containing the needed data. Node C subsequently sends the replicate command to nodes F and C. Node F is qualified thus it retrieves a portion of the file from the nearest nodes having the data (e.g. nodes B or C). Nodes C and /are not qualified thus they receive nothing. Node G is a terminating node because the rolled-up attribute of its branch does not satisfy the distribution criteria. This initial replication process continues until all the qualified nodes in SCDN are at least partially populated. In one or more embodiments, the same portion (e.g., blocks) of the large payload file is contained in at least one node of the SCDN. Preferably, a plurality of nodes maintains the same portion thereby creating redundancy and preventing loss of any portion of the large payload file when one or more nodes or storage volumes become unavailable. For example, when a storage volume (or device) becomes unavailable (i.e., lost), a DS at that station need not take any special action to recover contents of the damaged volume since the portions of large

payload files stored and hence lost in that volume are automatically downloaded from other network nodes upon demand to service a user request. The distribution servers also relay control information of a failed station to neighbors of the failed station to prevent improper termination of control commands.

Detail Description Paragraph:
[0121] The FDP uses the content provider's distribution criteria to direct the distribution of the large payload file in whole or in part to all nodes in the network meeting the provider's distribution criteria. A distribution request can start from any node in the tree, and traverses up and down the tree until it reaches a leaf node or arrives at a terminating node. For any node having the appropriate attributes, the file is partially downloaded from the nearest neighbors that meet specific performance criteria if those neighbors contain the portion of the file to be downloaded. The nearest neighbor when downloading content is not necessarily the nearest in the virtual tree but nearest in terms of distance. This prevents massive transfers from the node at which the file is initially uploaded. Moreover, the staging nature of the distribution prevents excessive demands on the network around the initial node (e.g., node B). By delivering smaller blocks and only a partial file this delivery method reduces network load. Additionally, because the distribution requests stop progressing through the SCDN when they arrive at a "terminating" node, the present invention prevents unnecessary distribution request packets from flooding the network.

Detail Description Paragraph:
[0140] A SCDN in accordance with an embodiment of the present invention is highly scalable. For example, when a new node is added to the SCDN, it downloads the initial content it needs by employing one of several different adaptable initialization processes. In one embodiment of the invention, an "Auto-initialization" process is used. When a node is added to an SCDN, it is given a set of attributes. In the auto-initialization process, as soon as the node is connected to the network, it issues an FDP "Learn" or similar request to all its neighbors. The node encodes its attributes in the learn request. The neighbors offer content, consistent with the new node's attributes, to it for downloading. The neighbors then pass on the new nodes learn request to all of their neighbors, which take similar action. Thus, the new node's learn request traverses the entire network and all the nodes in the network respond to the learn request if they have contents appropriate for the new node. The new node collects all the information, downloads the necessary initial contents, and is now a functioning element of the SCDN.

Detail Description Paragraph:
[0147] As in the earlier embodiments, the stations of SCDN 1400 are organized in a logical virtual tree structure in which each node in the tree has a set of attributes. Thus, each Station has an attribute set that is stored in the node and can be represented in any convenient data structure, e.g., the attribute set can be represented as an attribute bitmap. Furthermore, each Station (i.e., node) also contains a representation of the rolled up attribute set of each of the station's child-Stations. This representation is called the "Rolled Up Set of Attributes" and any convenient data structure can be used for it, e.g., a "Rolled Up Bitmap", which may be the defined as the "binary OR" combination of all rolled up attribute bitmaps from the child-Stations. The distribution servers within a Distribution Server Cluster use the attribute bitmap to distribute and route portions of large payload files and they use the aggregated rolled-up attribute bitmap to terminate unnecessary propagation of messages. One of the Stations in an SCDN is designated the "Central Station". The Central Station holds an attribute database table that matches text strings to bit positions, e.g., a reference table. Central Station 1420 is not necessarily a data repository for all content but may contain some or all the content.

Detail Description Paragraph:
[0218] There are multiple storage volumes in the shared Storage System 1530 (FIG.

15). These volumes contain a number of important databases illustrated in FIG. 20 such as: The Content Provider Data Table, File Distribution Criteria Table, the File Metadata Database, and the content block files. These databases are stored independently and may be mapped arbitrarily anywhere within shared Storage System 1530. The Content Provider Data Table and File Distribution Criteria Table also exist in each Content Management Server system. The tables in the Content Management Server only include the data of the content providers that are assigned to that CMS.

Detail Description Paragraph:
[0230] Distribution servers communicate with one another and the Content Management application in order to transfer large payload files in the SCDN. The DSs modify entries in the File Metadata Database when they add or remove blocks in the shared Storage System. A sequence server, which serves essentially as a lock manager, may be used to synchronize access to the file metadata database by multiple DSs, VFCS servers, storage managers, etc., to prevent possible race (e.g., conflict) condition. VFCS servers use information in the File Metadata Database to assemble and multiplex appropriate blocks into files for the Application Servers. The Storage Management Subsystem watches the available shared storage, the content provider's reserved storage, and the usage logs. It initiates the removal of less popular content to make room for more popular and new content when available storage is running low. It does this by instructing the DS to remove some of their associated blocks that are least likely to be used. It accesses the File Metadata Database to determine how many and which blocks it will request to be deleted.

Detail Description Paragraph:
[0281] The Content Management Server is a tool that acts as a bridge between the Content Management Client tool and the distribution server. It communicates with a DS in its assigned station using the FDP protocol (e.g., using the "put", "distribute", and "clean" commands). When the Content Management Client tool uploads a new file, the Content Management Server tool distributes (i.e., injects) the file into the SCDN via its assigned DS using FDP "put" and "distribute" packets. Content Management Server saves content provider information, and content distribution criteria in its database. While content is distributed to the SCDN, the DSs involved in processing the distribution request store information related to the files and their constituent portions in the File Distribution Criteria Database and the File Metadata Database. The files and their constituent blocks are stored in the content repository distributed in the SCDN, also by DS involved in processing the request. The Content Management Server uses the FDP "clean" packet to remove a file from an SCDN. When a DS in the SCDN receives a "clean" packet, it removes the relevant information from File Distribution Criteria Database, File Metadata Database, and the actual content blocks from the content repository.

First Hit

☐ Generate Collection

L5: Entry 1 of 7 File: PGPB Mar 6, 2003

DOCUMENT-IDENTIFIER: US 20030046369 A1
TITLE: Method and apparatus for initializing a new node in a network

Abstract Paragraph:
Large payload files are selectively partitioned in blocks and the blocks distributed to a plurality of distribution stations at the edge of the network qualified to have the data. Each qualified station decides how much and what portion of the content to save locally, based on information such as network location and environment, usage, popularity, and other distribution criteria defined by the content provider. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server, a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, a cluster of distribution servers at the distribution station may download the non-resident portions of the file as the application server is servicing the user. The download may be in parallel and usually from the least congested nodes. New nodes added to the network learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage devices from the nearest and least congested nodes without interrupting network operation. Each node manages its local storage and decides what content to prune based on information such as usage patterns.

Cross Reference to Related Applications Paragraph:
[0001] This application is a divisional of U.S. Application No. 09/681,644, filed on May 15, 2001, entitled "Method and Apparatus For Large Payload Distribution in a Network," which claims the benefit of U.S. Provisional Application No. 60/266,286, filed on Oct. 26, 2000, entitled "Large Payload Delivery Networks Having Integrated Content Management Services," the specification of which is herein incorporated by reference.

Summary of Invention Paragraph:
[0005] Network-based content delivery that relies on a single source to simultaneously distribute various types of information to multiple remote locations may, depending on the size of files being transferred, encounter network-loading problems around the server or the server itself may be over tasked. For example, since transferring a small file (e.g., a web-page) usually takes only a few seconds, the massive distribution of a small file from one source to thousands of destination locations may not create large impact on the network traffic near the source. Transferring a large file (i.e., a large payload), in contrast, can take tens of minutes to hours. If the distribution of such payloads relies on a single source, the network performance near the source, and the subsequent delivery of content, could degrade severely and become unacceptable.

Summary of Invention Paragraph:
[0007] The fast-paced expansion of the broadband industry has fueled the push for rich media (e.g., full length movies, video, or other types of multimedia data). Broadband technology brings high-speed connection capabilities for content delivery to remote users hence large payloads can be transferred faster. Also, broadband

technology makes it possible to send audio and/or video data using streaming media whereby the data is sent in streams for real-time playback, for example. Thus, the quality of rich media at the user's terminal, more than that of any other type of information, is now more dependent on the performance capabilities of the delivery technology. In order to minimize delivery delays, network congestion, and other related problems, some systems attempt to locate content on server systems that are located in close proximity to, i.e., a few hubs of connections away from the end-users. These server locations approximately define the concept known as the "edge" of the network. For example, the Internet service providers are in close proximity to the end-user thus may be regarded as being at the edge of the network. When servers are placed in such locations, the servers are said to be at the edge of the network. End-user systems that are configured to obtain content from network nodes located at the edge of the network are therefore beyond the edge of the network (a.k.a. last mile). However, it is important to note that systems located beyond the edge of the network are still coupled to the network and capable of communicating with the server computers located at the edge. Placing content at the edge of the network is advantageous because it can reduce the latency in servicing users located beyond the edge. Current approaches for delivering large payloads to the "edge" consist of mirroring or caching. These approaches and the limitations inherent in each approach will now be discussed in detail so as to give the reader an understanding of the advancements made by the invention.

Summary of Invention Paragraph:
[0035] Therefore, there is a need to address the cost, scalability, and load-balancing issues associated with large payload delivery to the edge of the network. However, before discussing the present invention, a general overview of how files are handled in different operating systems is presented.

Summary of Invention Paragraph:
[0041] An embodiment of the invention provides an improved mechanism for distributing large files throughout a computer network and delivering such files to an end-user system. When the invention is implemented it provides multiple users with a way to obtain access to large payload files without overburdening network resources. If, for example, a user wishes to download a large file such as a video file an embodiment of the invention provides a way to deliver that video file to the requesting user without putting a strain on the network. The system accomplishes this by breaking the large file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of large data files. Thus, the portions of large data file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations are referred to by those of ordinary skill in the art as the edge of the network.

Summary of Invention Paragraph:
[0043] In one embodiment of the invention, the system is optimized so that large payload files can be distributed across existing networks (including the Internet and corporate intranets) using a transport layer network overlay to push content to the edge of the network. Specifically, the embodiments of the invention improve large payload delivery performance, scalability, reliability, and availability.

Summary of Invention Paragraph:
[0044] As mentioned above, one embodiment of the invention breaks the large payload files into multiple portions. This may be accomplished by selectively partitioning the large payload file into blocks that are replicated and distributed to a plurality of distribution stations (a.k.a. nodes) at the edge of the network. Each distribution station is configured to determine how much of the content to save locally, based on information such as usage, popularity, etc. The content provider defines what distribution stations are qualified to function as distribution

stations and may also define other distribution criteria. Distribution stations in the network manage storage and transfer content (e.g., portions of large payload files) and other information to one another. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server (e.g., a streaming server), a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, the distribution stations may download the non-resident portions of the file as the application server is servicing the user. The download of the non-resident blocks may be in parallel and usually from the least congested nodes. The entire process is transparent to the user.

Summary of Invention Paragraph:
[0047] In one or more embodiments, new nodes may be added to the network without service interruption. As the new nodes are added, they learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage from the nearest and least congested nodes. Thus, a node could be added to the network and it would be up and running after self-initialization.

Summary of Invention Paragraph:
[0048] In one or more embodiments, the portions and amount of a large payload file maintained at each node depends on the available storage, popularity of the content, distribution criteria by the content provider, etc. Thus, least likely to be used blocks of a large payload file may be pruned (i.e., deleted from local storage) to make room for other highly desirable content. However, although the least likely to be used blocks of a file are pruned, the entire content of a large payload file may be maintained at a node in the scalable content delivery network, so long as the content provider wants the content to remain in the network.

Brief Description of Drawings Paragraph:
[0053] FIG. 5 is an illustration of a scalable content delivery network for delivering large payloads according to an embodiment of the present invention.

Brief Description of Drawings Paragraph:
[0061] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within the network of the present invention.

Detail Description Paragraph:
[0075] When the invention is implemented in accordance with one embodiment of the invention it provides end-user systems with a way to access large payload files without overburdening the network utilized by the end-user system to transmit data. In one embodiment of the invention, the system accomplishes this by breaking the large payload file into multiple portions and storing those portions in locations (e.g., nodes) distributed throughout the network. The portions stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of the large payload files. Thus, portions of the large payload file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations minimize the latency associated with delivering the file to the end-user system and are referred to herein as the edge of the network.

Detail Description Paragraph:
[0077] FIG. 5 provides a view of a scalable content delivery network (SCDN) for delivering large payloads according to an embodiment of the present invention. SCDN 500 may be a network such as the Internet which conceptually includes a network core 505 (i.e., the backbone), intermediate network segments 510 ranging "near" and "far" from the core, and network segments "far" from core 520-A through 520-C (collectively 520). "Near" and "far" relate to distance and are intended to

indicate relative path latencies (short or long, respectively) to the core, such latencies generally depend on the number of intermediate hubs (e.g., switches, routers, and the like) that are traversed to reach the high-speed backbones that form the core of the network and through which much of the network traffic is routed. Note that each intermediate hub may perform some limited processing, which adds latency, before forwarding the traffic to the next hub.

Detail Description Paragraph:
[0078] FIG. 5 shows a plurality of Content Provider Client (CPC) systems 530, a plurality of End-User Client (EUC) systems 550, and one or more Content Management Servers (CMS) 570, all located beyond Network Edge 501. In general, the content provider client 530 may be connected (or assigned) to a content management server 570, which in turn is connected to its assigned distribution center 540. A content provider uploads and/or manages large payload files in the SCDN 500 through its CPC 530. The EUC 550 provides the end-user access to files in SCDN 500. For example, EUC 550 may be a browser running on the end-user's local computer.

Detail Description Paragraph:
[0079] Network Edge 501 generally may be far from network core 505. However, the distance (i.e., path latency) between the core and the edge may not be uniform and may vary considerably for a given CPC or EUC. One embodiment of the present invention places a plurality of Distribution Centers (DC) 540A-540I for maintaining large payloads at the edge of the network thereby resolving the latency issue. Large payload content from a content provider is pushed from one distribution center to other distribution centers at the edge of the network. An end-user seeking access to a large payload is serviced (via an application server) from the nearest distribution center containing the desired content. By distributing content to the end-user (e.g., at EUC 550) via a plurality of Application Servers 560 and distribution centers 540 at the edge, path latency is minimized. Thus, large payload distribution involves obtaining a large payload file from a content provider and geographically placing such file at the distribution centers which are at or as close to the edge of the network as possible.

Detail Description Paragraph:
[0082] Content management server 570 may be connected to any node on the tree. Thus, although a content management server and a distribution center may not be collocated, the content management server gives the content provider a vehicle to upload large files (e.g., video) to the distribution centers. In one embodiment, the content management server is a computer that processes the content provider's large payload file for distribution in the network. In another embodiment, the content management server may, for example, be a subset of tools (e.g., machine independent objects) that allows upload of content to the network; thus, the tools may be shipped from a server to the content providers client's computer for processing and distribution of the large payload file in the network. After a content provider loads the large payload file into the content management server, the CMS may process the file and forward it to the distribution center.

Detail Description Paragraph:
[0085] The Virtual File Control System (VFCS) 702 is able to piece the original (large payload) file back together from the block files. As will be explained later, all the blocks of the large payload file need not be stored at one distribution center, however, the entire file is available within the SCDN. When an end user connects to application server 721 (e.g., a streaming server), the VFCS creates a virtual appearance that the entire file is available at that node. For example, assuming only fifteen percent of a two-gigabyte file is stored in storage 711-713, the VFCS makes streaming server 721 think that the entire two gigabytes is available at the location. Thus, streaming server 721 may start playing the file. As the file is being played, VFCS communicates with DS to locate and retrieve the remaining portions of the file from other nodes in the network.

Detail Description Paragraph:
[0110] To distribute a file, a content provider sets specific distribution criteria for that file. After the distribution server (DS) stores the uploaded large payload file as blocks, the content provider requests, through the content management server, that the DS distribute the file to other nodes in the SCDN, i.e., to push the content to the edge of the network. The distribution is in accordance with specific distribution criteria set by the content provider and may use the file distribution protocol (FDP) previously described. The distribution criteria may specify regions (e.g., Europe), specific nodes, and other information as desired by the content provider to control distribution of the content. For example, the distribution criteria may include information found in the nodes attribute or rolled up attribute bitmap.

Detail Description Paragraph:
[0111] The file distribution proceeds as follows: (1) The DS responds to the content provider's request to distribute a large payload file by sending a notification (i.e., a distribution request) to its neighbors to announce the existence and the distribution criteria of the file; (2) "Qualified" neighbors (i.e., those that meet the criteria) download several portions of the file during this initial distribution process; (3) The notification is then passed on from neighbor to neighbor, but not back to the neighbor from which the distribution request is received; (4) Each neighbor performs steps 2 and 3 until it encounters a leaf node or a "terminating" node. Thus, the distribution of the file in the network is done in stages.

Detail Description Paragraph:
[0113] FIG. 13 is an illustrative embodiment of the distribution of a large payload file within an SCDN. A content provider uploads a large payload file into the content management server (CMS) 570, which is connected to node B of the SCDN, using any content publishing and management software running on the content provider's client system (CPC) 530. The content provider also uploads the distribution criteria onto CMS 570. Content management server 570, as previously described, divides the uploaded file into track files and issues a command similar to the FDP "put" command for each track file to the distribution server located in node B. In other embodiments, the CMS may be connected to any node of the SCDN. At node B, the DS divides the track files into block files for local storage. The full copy of the file is shown at Node B as a filled in dot. The CMS then issues an FDP command of the type "distribute" to the distribution server at node B. In response to the distribute command, the DS issues a command to its neighboring nodes A, D, and E to replicate the content (e.g., using the "replicate" command of the FDP). Node D examines the replicate packet and decides its not supposed to have the content thus it passes the replicate command to its neighbor, node H. Nodes A, E, and H examine the replicate packet and decide they all match the distribution criteria (i.e., they are "qualified" nodes). When ready, nodes A, E, and H issue commands to retrieve a portion of the file from the nearest node (e.g., node B) in the SCDN. Nodes E and Hare leaf nodes thus they do not propagate the replicate command. However, node A is the root node with child nodes Band C. Node A may not send the replicate command back to node B, because it is the originating node. However, node A may send the replicate request to node C. Node C checks the distribution criteria and decides it's a qualified node therefore it retrieves a portion of the file from the nearest nodes (e.g., the nearest of nodes A, B, E, and H) containing the needed data. Node C subsequently sends the replicate command to nodes F and C. Node F is qualified thus it retrieves a portion of the file from the nearest nodes having the data (e.g. nodes B or C). Nodes G and I are not qualified thus they receive nothing. Node G is a terminating node because the rolled-up attribute of its branch does not satisfy the distribution criteria. This initial replication process continues until all the qualified nodes in SCDN are at least partially populated. In one or more embodiments, the same portion (e.g., blocks) of the large payload file is contained in at least one node of the SCDN. Preferably, a plurality of nodes maintains the same portion thereby creating redundancy and

preventing loss of any portion of the large payload file when one or more nodes or storage volumes become unavailable. For example, when a storage volume (or device) becomes unavailable (i.e., lost), a DS at that station need not take any special action to recover contents of the damaged volume since the portions of large payload files stored and hence lost in that volume are automatically downloaded from other network nodes upon demand to service a user request. The distribution servers also relay control information of a failed station to neighbors of the failed station to prevent improper termination of control commands.

Detail Description Paragraph:
[0115] The FDP uses the content provider's distribution criteria to direct the distribution of the large payload file in whole or in part to all nodes in the network meeting the provider's distribution criteria. A distribution request can start from any node in the tree, and traverses up and down the tree until it reaches a leaf node or arrives at a terminating node. For any node having the appropriate attributes, the file is partially downloaded from the nearest neighbors that meet specific performance criteria if those neighbors contain the portion of the file to be downloaded. The nearest neighbor when downloading content is not necessarily the nearest in the virtual tree but nearest in terms of distance. This prevents massive transfers from the node at which the file is initially uploaded. Moreover, the staging nature of the distribution prevents excessive demands on the network around the initial node (e.g., node B). By delivering smaller blocks and only a partial file this delivery method reduces network load. Additionally, because the distribution requests stop progressing through the SCDN when they arrive at a "terminating" node, the present invention prevents unnecessary distribution request packets from flooding the network.

Detail Description Paragraph:
[0132] A SCDN in accordance with an embodiment of the present invention is highly scalable. For example, when a new node is added to the SCDN, it downloads the initial content it needs by employing one of several different adaptable initialization processes. In one embodiment of the invention, an "Auto-initialization" process is used. When a node is added to an SCDN, it is given a set of attributes. In the auto-initialization process, as soon as the node is connected to the network, it issues an FDP "Learn" or similar request to all its neighbors. The node encodes its attributes in the learn request. The neighbors offer content, consistent with the new node's attributes, to it for downloading. The neighbors then pass on the new nodes learn request to all of their neighbors, which take similar action. Thus, the new node's learn request traverses the entire network and all the nodes in the network respond to the learn request if they have contents appropriate for the new node. The new node collects all the information, downloads the necessary initial contents, and is now a functioning element of the SCDN.

Detail Description Paragraph:
[0139] As in the earlier embodiments, the stations of SCDN 1400 are organized in a logical virtual tree structure in which each node in the tree has a set of attributes. Thus, each Station has an attribute set that is stored in the node and can be represented in any convenient data structure, e.g., the attribute set can be represented as an attribute bitmap. Furthermore, each Station (i.e., node) also contains a representation of the rolled up attribute set of each of the station's child-Stations. This representation is called the "Rolled Up Set of Attributes" and any convenient data structure can be used for it, e.g., a "Rolled Up Bitmap", which may be the defined as the "binary OR" combination of all rolled up attribute bitmaps from the child-Stations. The distribution servers within a Distribution Server Cluster use the attribute bitmap to distribute and route portions of large payload files and they use the aggregated rolled-up attribute bitmap to terminate unnecessary propagation of messages. One of the Stations in an SCDN is designated the "Central Station". The Central Station holds an attribute database table that matches text strings to bit positions, e.g., a reference table. Central Station 1420 is not necessarily a data repository for all content but may contain some or

all the content.

Detail Description Paragraph:
[0209] There are multiple storage volumes in the shared Storage System 1530 (FIG. 15). These volumes contain a number of important databases illustrated in FIG. 20 such as: The Content Provider Data Table, File Distribution Criteria Table, the File Metadata Database, and the content block files. These databases are stored independently and may be mapped arbitrarily anywhere within shared Storage System 1530. The Content Provider Data Table and File Distribution Criteria Table also exist in each Content Management Server system. The tables in the Content Management Server only include the data of the content providers that are assigned to that CMS. The Content Provider Data Table at each station includes the information of all the content providers of the SCDN, and the File Distribution Criteria Table includes only the media files that are replicated to the station.

Detail Description Paragraph:
[0218] Distribution servers communicate with one another and the Content Management application in order to transfer large payload files in the SCDN. The DSs modify entries in the File Metadata Database when they add or remove blocks in the shared Storage System. A sequence server, which serves essentially as a lock manager, may be used to synchronize access to the file metadata database by multiple DSs, VFCS servers, storage managers, etc., to prevent possible race (e.g., conflict) condition. VFCS servers use information in the File Metadata Database to assemble and multiplex appropriate blocks into files for the Application Servers. The Storage Management Subsystem watches the available shared storage, the content provider's reserved storage, and the usage logs. It initiates the removal of less popular content to make room for more popular and new content when available storage is running low. It does this by instructing the DS to remove some of their associated blocks that are least likely to be used. It accesses the File Metadata Database to determine how many and which blocks it will request to be deleted.

Detail Description Paragraph:
[0267] The Content Management Server is a tool that acts as a bridge between the Content Management Client tool and the distribution server. It communicates with a DS in its assigned station using the FDP protocol (e.g., using the "put", "distribute", and "clean" commands). When the Content Management Client tool uploads a new file, the Content Management Server tool distributes (i.e., injects) the file into the SCDN via its assigned DS using FDP "put" and "distribute" packets. Content Management Server saves content provider information, and content distribution criteria in its database. While content is distributed to the SCDN, the DSs involved in processing the distribution request store information related to the files and their constituent portions in the File Distribution Criteria Database and the File Metadata Database. The files and their constituent blocks are stored in the content repository distributed in the SCDN, also by DS involved in processing the request. The Content Management Server uses the FDP "clean" packet to remove a file from an SCDN. When a DS in the SCDN receives a "clean" packet, it removes the relevant information from File Distribution Criteria Database, File Metadata Database, and the actual content blocks from the content repository.

CLAIMS:

1. A method for initializing a new node in a network comprising: adding a new node to a network having a plurality of nodes; sending a query automatically to said plurality of nodes to determine what contents to download; receiving replies to said query from a subset of said plurality of nodes having said contents for said new node; downloading desired portions of said contents from said subset of said plurality of nodes having said contents.

15. A method for initializing a new node in a network comprising: adding a new node to a network having a plurality of nodes, wherein said plurality of nodes is

arranged in the form of a virtual tree and said new node is a node of said tree, each node of said tree having a set of attributes and a set of rolled up attributes for identification; sending a query from said new node to said plurality of nodes to determine what contents to download, said contents being stored as block files in one or more nodes of said network; receiving replies to said query from a subset of said plurality of nodes having said contents for said new node; downloading desired portions of said contents from said subset of said plurality of nodes having said contents.

16. A method for initializing a new node in a network comprising: adding a new node to a network having a plurality of nodes, wherein said plurality of nodes is arranged in the form of a virtual tree and said new node is a node of said tree, each node of said tree having a set of attributes and a set of rolled up attributes for identification; sending a query automatically from said new node to said plurality of nodes to determine what contents to download, said content being stored as block files in one or more nodes of said network, said query comprising said set of attributes and said set of rolled up attributes for said new node; receiving replies to said query from a subset of said plurality of nodes having said contents for said new node, wherein each of said replies identifies what subset of said block files is available in a replying node and performance characteristics of said replying node; downloading desired subsets of said block files from said replying nodes that are least congested.

17. A computer program product comprising: a computer usable medium comprising computer readable code for initializing a new node in a network, said computer readable program code configured to: add a new node to a network having a plurality of nodes; send a query automatically to said plurality of nodes to determine what content said new node should have; receive replies to said query from a subset of said plurality of nodes having said content for said new node; download a desired portion of said content from said subset of said plurality of nodes having said content.

31. An apparatus for initializing a new node in a network comprising: a network having a plurality of nodes, each of said plurality of nodes having one or more distribution servers in a distribution server cluster, said plurality of nodes having one or more content for distribution in said network; a new node added to said network, said new node sending a query automatically to said plurality of nodes to determine what contents to download, said new node receiving replies to said query from a subset of said plurality of nodes having said contents for said new node, said one or more distribution servers in said distribution server cluster in said new node downloading desired portions of said contents from said subset of said plurality of nodes having said contents.

First Hit Fwd Refs

Generate Collection

L10: Entry 16 of 36 File: USPT Apr 22, 2003

DOCUMENT-IDENTIFIER: US 6553413 B1
TITLE: Content delivery network using edge-of-network servers for providing content delivery to a set of participating content providers

Application Filing Date (1):
20000628

Brief Summary Text (17):
A further feature of the present invention is the ability to distribute and manage content over a large network without disrupting the Content Provider's direct relationship with the end user.

Detailed Description Text (46):
Performance for long downloads can also be improved by dynamically changing the server to which a client is connected based on changing network conditions. This is especially helpful for audio and video downloads (where the connections can be long and where quality is especially important). In such cases, the user can be directed to an alternate server in mid-stream. The control structure for redirecting the client can be similar to that described above, but it can also include software that is placed in the client's browser or media player. The software monitors the performance of the client's connection and perhaps the status of the network as well. If it is deemed that the client's connection can be improved by changing the server, then the system directs the client to a new server for the rest of the connection.

Detailed Description Text (64):
Competing solutions are not scalable to more than a small number of sites. For example, solutions based on mirroring are typically used in connection with at most three or four sites. The barriers to scaling include the expense of replicating the entire site, the cost of replicating computing resources at all nodes, and the complexity of supporting the widely varying software packages that Content Providers use on their servers.

Detailed Description Text (65):
The unique system architecture of the present invention is scaleable to hundreds, thousands or even millions of nodes. Servers in the hosting network can malfunction or crash and the system's overall function is not affected. The global hosting framework makes efficient use of resources; servers and client software do not need to be replicated at every node because only the hosting server runs at each node. In addition, the global hosting server is designed to run on standard simple hardware that is not required to be highly fault tolerant.

Detailed Description Text (86):
Further, as used herein, a Web "client" should be broadly construed to mean any computer or component thereof directly or indirectly connected or connectable in any known or later-developed manner to a computer network, such as the Internet. The term Web "server" should also be broadly construed to mean a computer, computer platform, an adjunct to a computer or platform, or any component thereof. Of course, a "client" should be broadly construed to mean one who requests or gets the file, and "server" is the entity which downloads the file.

☐ Generate Collection

L10: Entry 15 of 36 File: USPT Sep 23, 2003

DOCUMENT-IDENTIFIER: US 6625643 B1
TITLE: System and method for resource management on a data network

Application Filing Date (1):
19991113

Brief Summary Text (4):
The Internet is a loose network of connected computers spread throughout the world. A message can be sent from any computer on the Internet to any other by specifying a destination address and passing the message from computer to computer via a series of "hops." Each computer, router, or "node" on the Internet has a unique Internet address. When an intermediate computer or router receives a message in transit, the computer checks the intended destination of the message and passes it along accordingly.

Brief Summary Text (12):
Traditionally, outside of the Internet, the primary method for communicating electronically with a substantial number of customers or users has been broadcasting. Radio, television, and other media all use various forms of broadcasting. Although it is possible to reach large numbers of people this way, it is difficult to regulate distribution and receipt of the content.

Detailed Description Text (301):
In one embodiment, the publisher must begin a broadcast from within PowerPoint. This enables PowerVU to begin the downloading process of the broadcast from the publisher's local system to the PowerVU server and ensures PowerPoint and the broadcast are ready to go.

First Hit Fwd Refs

☐ Generate Collection

L8: Entry 3 of 17 File: USPT Sep 30, 2003

DOCUMENT-IDENTIFIER: US 6628655 B1
TITLE: Method of self-learning for the switching nodes of a data transmission network

Abstract Text (1):
Method of self-learning for a switching node in a data transmission network (10) wherein Internet Protocol (IP) data frames are transmitted between a sending unit (20 or 24) and a receiving unit (22 or 26) via an ingress switching node (12) linked to an egress switching node (16 or 18) by a plurality of intermediate switching nodes (14) interconnected by trunks. The method consists principally in generating, in the input port/trunk interface of the switching node, a switch header including a source identification field, a destination identification field and a temporary label identifying the flow of data in response to its first data frame, this first switch header being concatenated to said data frame before being transmitted to the router interface of the switching node via its switch engine when the data frame does not belong to a known flow of data and the router has to define the routing of the flow of data. An update message is sent back from the output trunk/port interface to the input port/trunk interface to update the latter with the destination identification and with a switching label replacing the temporary label associated with the flow of data, in order to transmit the following frames of the flow of data directly to the trunk/port interface by using this switching label.

Application Filing Date (1):
20000124

Brief Summary Text (2):
The present invention relates generally to the switching of the data packets in each switching node of a data transmission network supporting the Internet Protocol (IP) and particularly to a method of self-learning for the switching nodes of such a data transmission network.

First Hit Fwd Refs

☐ Generate Collection

L3: Entry 1 of 2 File: USPT Nov 25, 2003

DOCUMENT-IDENTIFIER: US 6654759 B1
TITLE: Method for access via various protocols to objects in a tree representing at least one system resource

Application Filing Date (1):
20001127

Detailed Description Text (22):
The result of the implementation of the method just described is that an object (83 for example) is visible in two different ways in two different naming spaces. First of all, it is visible in the space 86c of the target object 81 through the naming attribute, and additionally through the class of this object. In fact, the object 81 being a father object, it knows some of the attributes of its sons, including the naming attribute and the class. Among the (father) target objects and the son objects, it is possible to have objects that are different in terms of semantics, i.e., not necessarily of the same class. Secondly, it is visible in the naming space 86b of the object itself 83. The space 86b makes it possible to learn all of the attributes of the object 83 using a request based on the corresponding interface layer SPI 47b, defined from the "factory"--"URL" attributes of the associated context Cb.

First Hit

Hit List 1 of 36 File: PGPB Dec 25, 2003

DOCUMENT-IDENTIFIER: US 20030237016 A1

TITLE: System and apparatus for accelerating content delivery throughout networks

Application Filing Date:
20010301

Summary of Invention Paragraph:
[0007] In the world of the Internet, most service providers are interested in accommodating some number of `hits` per minute. Accommodating some number of `hits` translates generally into forming a corresponding number of TCP/IP connections and downloading at least one file via HTTP or FTP. Where traditional architectures are limited in their ability to handle increasing numbers of TCP/IP connections due to the software overhead involved in handling each session's state information, the service providers must generally calculate the number of `hits` per minute that a single server is capable of handling, and then calculate the number of servers generally required to be running in parallel to handle a projected, aggregate load. Such connection limitations and disadvantages may be experienced in both wireline and wireless content serving.

Summary of Invention Paragraph:
[0009] The majority of the traffic generated from web hosting sites may be characterized as asymmetric, favoring movement from web servers to clients, and involving the transfer of static content. Approximately 70% of HTTP requests for data are for static content. FTP, the primary protocol used for file transfers ranging from MP3 files to software upgrades, is also responsible for transferring large volumes of static content. In addition to file downloads, newer protocols exist which have been designed to download `streams` of static content such as video.

Summary of Invention Paragraph:
[0010] A content delivery solution which does not posses the drawbacks experienced with traditional server farms involves employing a content router which may be used to offload storage reads from a host server's CPU (central processing unit) and I/O sub-system (Input/Output). Such a configuration enables virtually unlimited bandwidth scalability without additional CPU processors. In essence, the content router serves, at least in part, as a uni-directional content transport network appliance that accesses content from storage and routes it to requesting IP (Internet Protocol) addresses over a network. When deployed in conjunction with a conventional server responsible for storage writes, network management, and system administration, the flexibility of the general-purpose computer is maintained while the reliability of a network appliance to access static content is leveraged. Applications that may benefit from such a content router include the aforementioned file downloading, static HTTP content serving and streaming media, as well as web caching and other applications with intensive read operations.

Summary of Invention Paragraph:
[0011] Accordingly, the present invention provides a system for rapidly delivering large volumes of content from storage. The system preferably includes at least one content router having at least one network processor, memory operably coupled to the at least one network processor and at least one routing switch operably coupled

to the network processor. In addition, the content router preferably includes a LAN interface operably coupled to the routing switch that is preferably operable to communicate with a local area network, a WAN interface operably coupled to the routing switch that is preferably operable to communicate with a wide area network and a SAN (storage area network) interface operably coupled to the routing switch that is preferably operable to communicate with a SAN. A program of instructions storable in the memory and executable in the processor is also included and is preferably operable to interrogate packets received through the WAN interface and to instruct the routing switch to switch the packets between the LAN, WAN and SAN interfaces based upon the results of the interrogation. At least one storage device coupled to the SAN interface may also be included in the system.

Summary of Invention Paragraph:
[0018] The present invention further provides technical advantages of autonomous content streaming, infinite up-scaling of download throughput bandwidth and low-overhead system administration that scales logarithmically with throughput bandwidth as well as seamlessly integrating with standard network and system management software packages.

Detail Description Paragraph:
[0130] In the case of a layer 3 or layer 2 switch replacing load balancing web switch 1115, content router 200 preferably handles all load balancing and switching to nodes within the content router 200 cluster, including that traffic which needs to be switched to server 215. In this scenario, there may be a primary content router 200 in the cluster that is operable to load balance TCP/IP connection setups and application layer protocols across the remaining content routers 200 and servers 215 in the cluster.